THE FEASIBILITY OF CROSS-VALIDATION AS A PARAMETER PREDICTOR FOR THE ITERATIVE UNFOLD METHOD

THESIS

Dennis J. Miller
First Lieutenant, USAF

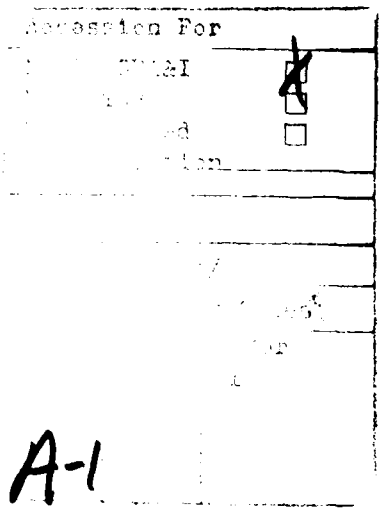
AFIT/GNE/ENP/90M-5

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GNE/ENP/90M-5



THE FEASIBILITY OF CROSS-VALIDATION AS A PARAMETER PREDICTOR FOR THE ITERATIVE UNFOLD METHOD

THESIS

Dennis J. Miller
First Lieutenant, USAF

AFIT/GNE/ENP/90M-5

AFIT/GNE/ENP/90M-5

THE FEASIBILITY OF CROSS-VALIDATION AS A PARAMETER PREDICTOR FOR THE ITERATIVE UNFOLD METHOD

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Nuclear Engineering (Effects)

Dennis J. Miller, B.S.
First Lieutenant, USAF

March 1989

Preface

The purpose of this study was to test the applicability of the method of cross-validation for optimizing the parameters used in the iterative unfold method. Specifically, the goal was to determine the feasibility of cross-validation as a means of predicting the optimum stopping point for the iterative unfold process. Users of the iterative unfold process have a need for a theoretically based approach that predicts when to stop the iterative process. During this study, a computer code was developed. The code consisted of an iterative unfold technique coupled with cross-validation. Test cases were constructed to assess cross-validation's ability to predict a meaningful stopping point for the iterative unfold process.

Guidance and support for this effort was received from many sources. First, I would like to thank my thesis advisor LCDR Kirk A. Mathews. His expertise in the unfold area proved invaluable. I would also like to thank my thesis committee; Maj. Cipperly and Capt. Sabochick for their assistance in the preparation of the final copy of this thesis. Next, I would like to thank Capt. Michael Carter and Capt. Russell Daniel. Their work on this problem provided the crucial starting point for this study. And finally, I would like to thank my wife, Laurie, and sons, Jeremy and Christopher for their patience and support.

Dennis J. Miller

Table of Contents

List of Tables

List of Figures

Abstract

Different applications of the Fredholm integral equation appear in varied fields of study. An application of particular interest to the Air Force arises when attempting to determine pulsed radiation spectra using measured data from underground nuclear effects simulations.

The iterative unfold technique provides a means for approximating a solution to a system of Fredholm integral equations. This method consists of modifying a guess spectrum through a fit process using data collected from experiment. Next, the unfolded spectrum is smoothed to reduce undesirable artifacts that result from the fitting process. Finally, the entire iterative process is then repeated as necessary to provide an approximation to the exact incident spectrum. Presently, the iterative unfold method lacks an independent measure of how well the unfolded spectrum approximates the exact spectrum. Consequently, user judgement is necessary, resulting in possible data overfitting. Cross-validation is a method which selectively partitions the measured data into subsets. The subsets of data are used to predict omitted data. A cross-validatory loss statistic can be formulated and minimized to predict the optimum stopping point for the iterative unfold process.

The iterative unfold technique was implemented into a computer program. Cross-validation was introduced to determine the cross-validatory loss statistic. The loss statistic was minimized by varying the number of iterations of the unfold routine over a wide range. Test cases were developed to compare the cross-validatory loss statistic with a χ^2 formulated using ideal signals.

Two basic test cases were studied. In these two cases, the guessed spectrum, the simulated noise set, and implementation of the iterative unfold technique were varied. Cross-validation provided an estimate for the stopping point of the unfold routine in each case. Using the smooth algorithm in tandem with the fit algorithm produced an overall better fit than using the fit algorithm alone. When the guessed spectrum was set equal to the exact spectrum; cross-validation predicted no modification to the guessed spectrum, while judgement alone tended to overfit data. Stoppage based on the value of a χ^2 test statistic proved highly problem dependent. In contrast, cross-validation averaged within one iteration of the optimum regardless of the problem parameters.

Executive Summary

Introduction

Approximate solutions to the Fredholm integral equation are needed in varied areas of study. The Fredholm integral equation is given by Equation (1). An application of the Fredholm integral equation of particular interest to the Air Force and arises from the measurement of pulsed radiation emitted from nuclear weapon detonations. These detonations are used to simulate weapons effects. The detection and measurement of the incident pulsed radiation is accomplished by using a set of detection systems with overlapping responses. The detections systems generate signals which are used in the deconvolution process.

The exact incident spectrum obtained from the weapons effects simulation is unknown. The approximation of this spectrum is the goal of the deconvolution process. Since there are an infinite number of spectra which generate the same detector output, the problem of spectrum deconvolution is ill-posed. Additionally, sources of uncertainty such as measurement and calibration error make the measured signals non-ideal. Consequently, approximation of the exact incident spectrum may be a formidable task.

One common approach to approximating the exact spectrum is called the iterative unfold process. This process consists of modification of a guessed or trial spectrum by a two step process. First, the guess spectrum is modified through

fitting. The fit routine is a scaling process which takes advantage of the knowledge gained from the measured signals coupled with instrument response characteristics. Second, the fitted spectrum is smoothed through an averaging process. The smoothing is accomplished to suppress undesirable spectral artifacts that result from scaling discontinuities that occur during the fitting process. Once the two step spectrum modification is complete, an unfolded approximation to the exact spectrum exists. This two step process is then repeated as necessary, using the unfolded spectrum as the new guessed spectrum. Typically, a χ^2 test statistic is implemented to judge when to stop the iteration process. Since the ideal signals are unknown, χ^2 is formulated using the measured signals which contain noise. However, the noise is an artifact of the experiment and not the incident spectrum. The fit process scales the guessed spectrum to the measured signals which forces χ^2 to decrease monotonically as the number of iterations increase. Consequently, χ^2 is measure of the fit error to a hypothetical spectrum which contains noise.

Cross-validation is a statistical method which selectively partitions the measurement data into subsets. The data subsets are used to form a prediction for the omitted data. Using this technique, a test statistic can be formulated which is a more independent measure of the unfolded spectrums ability to predict the exact incident spectrum than

χ^2. The purpose of this study is to explore the use of cross-validation to predict when to stop the iterative unfold process.

Theoretical Development

For this study two types of detector response functions were used. They were the open (filter) and closed (filter-fluorescer) systems. These functions are given by Equations (3) and (4), respectively. Both functions are asymmetric response functions.

As stated above, the iterative unfold process consists of fit and smooth routines. The fit process is given by Equation (14). The fitting is basically a scaling process. The spectrum at each location is scaled by the measured-to-unfolded ratio of signals. The ratio is, in turn, multiplied by a predetermined instrument weighting. The product of instrument weighting and measured-to-predicted ratio are summed over all instruments to determine the scaling factor. Once the spectrum has been fit, a smoother is used.

The smoothing process is given by Equation (15). The smoothing process averages out discontinuities resulting from the fit routine. It consists of a smoothing kernel, $f(E,E')$, which is multiplied by the fitted spectrum and integrated over all energies. The result is renormalized to the integral of the smoothing kernel. In this study, the smoothing kernel consisted of a three bin uniform distribution typical of that used in the industry.

After each iteration, the two step process returns an unfolded spectrum which is used as an approximation to the exact spectrum. With repeated iterations resulting in further spectrum modification, the question is when to stop iterating.

The cross-validatory approach to parameter estimation consisted of creating data subsets by omitting each instrument, one at a time. Since seventeen instrument were used for this study, seventeen data subsets were created. The data subsets were used to generate an unfolded spectrum. The unfolded spectrum was used to generate unfolded signals which were compared to the measured signals of the instrument that was omitted. This lead to formulation of a cross-validatory loss statistic. This statistic is given in Equation (21). The cross-validatory test statistic was minimized by varying the number of iterations over a wide range. The iteration number where the cross-validatory loss statistic reached its minimum was used as the optimum iteration number.

Results and Discussion

Two basic test cases were studied. The first case used a one temperature Planckian as the exact incident spectrum. All cases using this one temperature Planckian were prefixed TC1. The second case used a two temperature Planckian as the incident spectrum. Similarly, all cases using the two temperature Planckian as the incident spectrum were prefixed TC2. In each case the guessed spectrum was varied. In addition, the simulated noise set and the method of implemen-

tation (fit only or fit/smooth) of the iterative unfold process was varied. The results for test cases 1 and 2 with the first simulated noise set are contained in section IV. A second noise set was simulated for TC1 and TC2. The results are tabulated in Appendix A and B, respectively.

Analysis consisted of comparison of the absolute error of the unfolded spectrum for different stopping criteria. Several test statistics such as spectral norms were calculated. The results of the test cases are summarized in Tables 15 and 16.

Summary and Conclusions

From the test cases a series of observations were made. First, for each incident spectra χ^2_f exhibited a deeper minimum when the iterative unfold process used the fit/smooth routines in tandem over the fit routine alone. This indicated that smoothing the spectrum at each step allowed for a better overall fit. Second, guessing a spectrum equal to the incident (exact) spectrum lead to cross-validatory predictions of no modifications to the guessed spectrum. In contrast, using judgement based on χ^2 lead to overfitting of the data as a result of noise. Stoppage based on any predetermined value of χ^2 tended to be problem dependent. In other words, varying parameters of the problem tended to change the optimum value of χ^2. In contrast, cross-validation was less sensitive to parameter changes. Cross-validation averaged within one iteration of the optimum (minimum absolute error of the unfolded spectrum).

Recommendations

Based on the results of this study, three recommendations for further study are made. First, continue the work in the area of cross-validation to predict other parameters in the iterative unfold technique. Second, this study should be continued, exploring the use of cross-validation to predict parameters of the basis function method for deconvolution. Finally, the use of generalized cross-validation should be thoroughly explored as a "stand alone" means for spectrum deconvolution.

THE FEASIBILITY OF CROSS-VALIDATION AS A PARAMETER PREDICTOR FOR THE ITERATIVE UNFOLD METHOD

I. Introduction

Background

Fredholm integral equations of the first kind arise in many varied areas of study, including acoustics, optics, superconductivity, and aerodynamics. A particular application of interest to the Air Force arises from the measurement of pulsed radiation emitted from nuclear weapon detonations. These measurements are used to predict nuclear weapons effects.

The Fredholm integral equation of the first kind is given by

$$Y = \int_a^b R(E)S(E)dE \tag{1}$$

where

Y - the detector output signal

$R(E)$ - the detector response function

$S(E)dE$ - the radiation spectrum

Currently, detection of pulsed radiation is accomplished by using a set of overlapping response detection systems cov-

ering the energy region of interest. These detectors generate a set of measured signals. Each of the measured signals can be represented by

$$Y_i^M = \int_0^\infty R_i(E) S_E(E) dE \tag{2}$$

where

Y_i^M- the measured signal of the ith detector

$S_E(E)$- the exact or actual spectrum emitted

$R_i(E)$- the actual response of the ith detector

Since the number of detectors used during an experiment are finite and detector response functions are not ideal (i.e. rectangular shaped with negligible width), only a limited resolution of the exact spectrum can be achieved. Additionally, neither the exact spectrum delivered by a nuclear device detonation, nor the exact response functions of the detectors can be determined. Furthermore, error contributions from sources such as transmission, recording, and calibration add further complexity to the problem.

Problem

Since there are an infinite number of spectra which give the same detector output, deconvolution of the Fredholm equation is ill-posed.

Two common approaches are used to approximate the exact

spectrum implicitly given in Equation (1). These techniques are the iterative method (1:2-3, 2) and the basis function method (1:2-4). Each of these methods have has its own inherent strengths and weaknesses.

The iterative unfold technique folds a trial or guessed spectrum with calibrated response functions to generate signals. The trial spectrum is modified through a fitting procedure to produce a spectrum that generates fitted signals that are closer to the measured signals. Next, the spectrum is smoothed to ensure continuity. Once the fit and smooth procedures are accomplished, the entire process is repeated using the newly unfolded spectrum as the trial spectrum for the next iteration.

Two problems arise in the application of the iterative unfold method. First, the unfolded spectrum is prediction dependent and is a function of the type of fitting procedure used. Second, the iterative unfold method lacks an independent means of determining how well the unfolded spectrum approximates the incident spectrum. Most adaptations of iterative method use a test statistic to compare the measured and unfolded signals. If this statistic is allowed to become arbitrarily small, then the unfolded spectrum may become overly dependent on random noise in the signals. As a result of noise contributions, a trade off exists as to how small the test statistic can become and still reflect the global trend of the data and not the detail of random noise. The purpose of this study is to explore the use of a technique

called cross-validation to estimate the optimum stopping point for the iterative unfold process, and as a result, minimize the development of undesirable spectral artifacts resulting from noise contributions. Cross-Validation will be discussed in detail in Section II.

General Approach

The approach used in this study was to simulate the data that would normally be available to the analyst. These data are usually the predicted spectrum, the measured-to-predicted ratios, and the standard deviations of these ratios. Additionally, the exact spectrum is assumed known to perform necessary evaluation of the techniques employed.

Once the input data are constructed by folding the chosen exact spectrum with the response functions and adding some random noise (measurement error), an iterative unfold scheme similar to that implemented by Carter (2) is used to create an unfolded approximation to the exact spectrum. Normally, a χ^2 test statistic is used to stop the iterative process at some approximation to the exact spectrum. But, the exact spectrum is unknown and the measurements are used in constructing the unfolded spectrum, consequently, χ^2 does not give an independent measure of performance. As a result, cross-validation is implemented to partition the data into subsets in an attempt to determine a test statistic that will independently predict an iteration stopping point. Comparisons will be made between the cross-validatory test statistic, χ^2, and common spectral norms.

Scope

The iterative unfold method is used during this study to give an approximate solution to equation (1). This method will be modified to incorporate a cross-validatory test statistic to assess the methods performance, in lieu of the χ^2 test statistic. Since the use of experimental data implies an unknown exact incident spectrum, this study was limited to simulated data. As a result, a set of synthetic test cases were developed on an energy range from 1 E_0^1 to 64 E_0^1, where E_0^1 is an arbitrary energy unit.

Assumptions

In order to take advantage of previous work by Daniel (1) and Carter (2) similar assumptions were made. They are as follows:

1. There are seventeen detectors. Eleven of which are "closed" and six are "open" response detectors (discussed in Section II).

2. The resolving power of the detectors, $\Delta E/E$, is approximately 1.5.

3. The response functions are exact. (Recording, Transmission, and Calibration Errors will be discussed in Section II.)

Sequence of Presentation

Section II presents a detailed discussion of the iterative unfold method, potential sources of error, and cross-validation. Section III provides a detailed description of

the implementation of the theory contained in Section II. The results and discussion are presented in Section IV. Section V summarizes the results of this study. Recommendations for further study are suggested in Section VI.

II. Theoretical Development

Introduction

Section I introduced the concepts of iterative unfold and cross-validation. Section I also established the assumptions and goals of this study. This section provides the theoretical development of both the iterative unfold method and parameter estimation by cross-validation. The following topics are discussed: the types of detection systems and their associated response functions, definition of spectra and signals, the iterative unfold process, potential sources of error, and the cross-validatory approach to parameter estimation.

Response Functions

A detector response function is the normalized output of the detector system due to incident radiation at a particular energy E. Due to errors resulting from the measurement process, the exact response functions are not known to complete certainty. However, for the purposes of this study these response functions are assumed to be equal to the calibrated response functions. As stated in Section I, seventeen detector systems were used for this study. Two different types of detector systems were implemented. These systems are the open (fluorescer) and the closed (filter-fluorescer) systems. These response functions were derived by Daniels (1:A-1).

A fluorescer detection system consists of a radiation pipe containing a fluorescer material. Photons below the photoelectric k-edge of the fluorescer do not react and flow out of the pipe giving no signal. Photons above the k-edge of the fluorescer react to cause a cascade of x-ray fluorescence, but do so with less probability as their energies increase. The thickness and type of fluorescer material can be changed to achieve the desired sensitivity. Figure 1 is a sketch of a typical fluorescer detection system. Fluorescent radiation is directed by collimators to the detector. Detectors generally consist of solid state or x-ray diodes which convert the photon energy into a voltage or current signal. The response function for an ideal open detection system is characterized by zero response at energies below the fluorescer k-edge, and exponentially-decreasing responses at increasing energies above the k-edge. The open response function used here is

$$R_i^{open}(E) = \begin{cases} 0 & E < E_i^0 \\ \left(\frac{1}{E}\right)\left\{1 - \exp\left[-3\left(\frac{E_i^0}{E}\right)^3\right]\right\} & E \geq E_i^0 \end{cases} \quad (3)$$

where

$R_i^{open}(E)$ - the open function of the i^{th} detector

E_i^0 - the k-edge of the fluorescer for the i^{th} detector

E - Energy

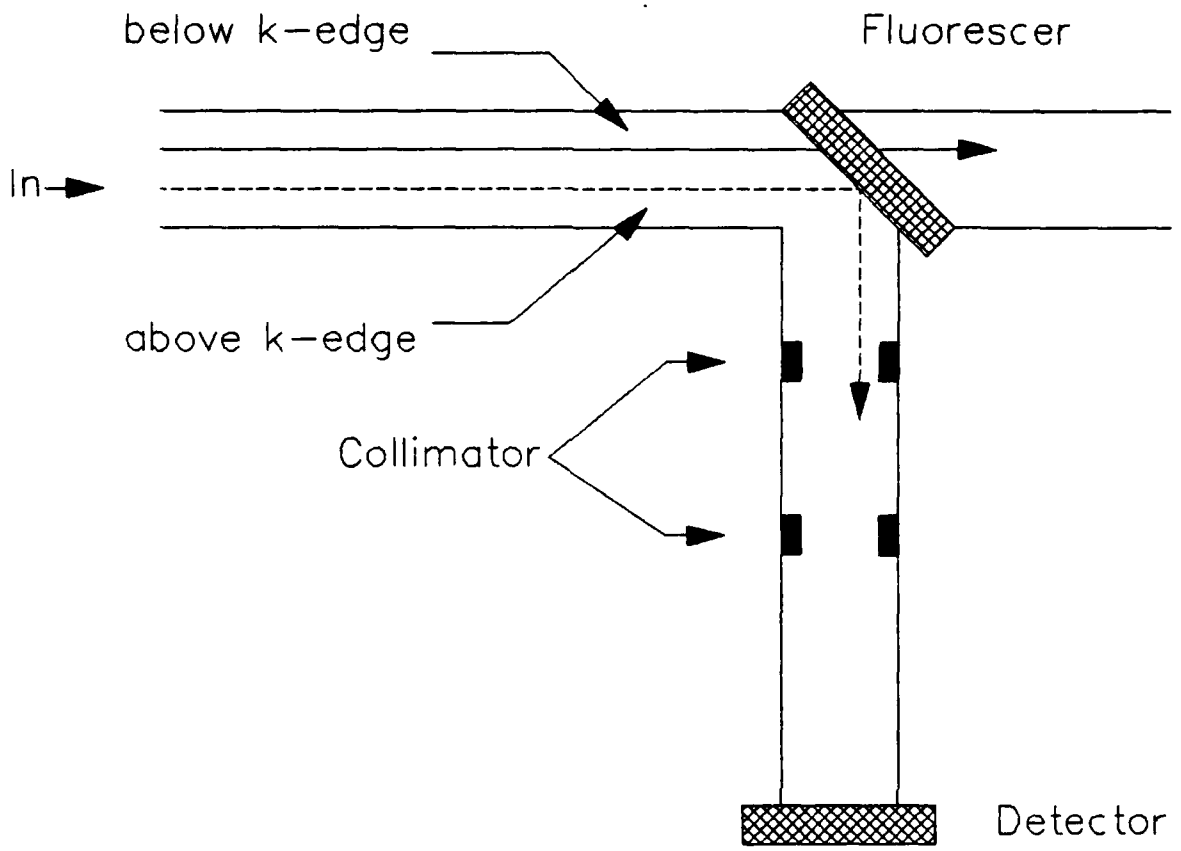


Figure 1: Typical Fluorescer Detection System

Figure 2 is an example of a typical open response function. In this example, the detector system has a k-edge of 2 E_0^1, where E_0^1 is defined as the lowest k-edge energy of the entire detection system.

The second type of detection system is the "closed" detection system. The closed detector system consists of a filter and a fluorescer. A typical filter-fluorescer system is similar in design to the fluorescer system, except the inbound pipe is obstructed by a filter material. Photons above the filter k-edge fluoresce and degrade in energy.

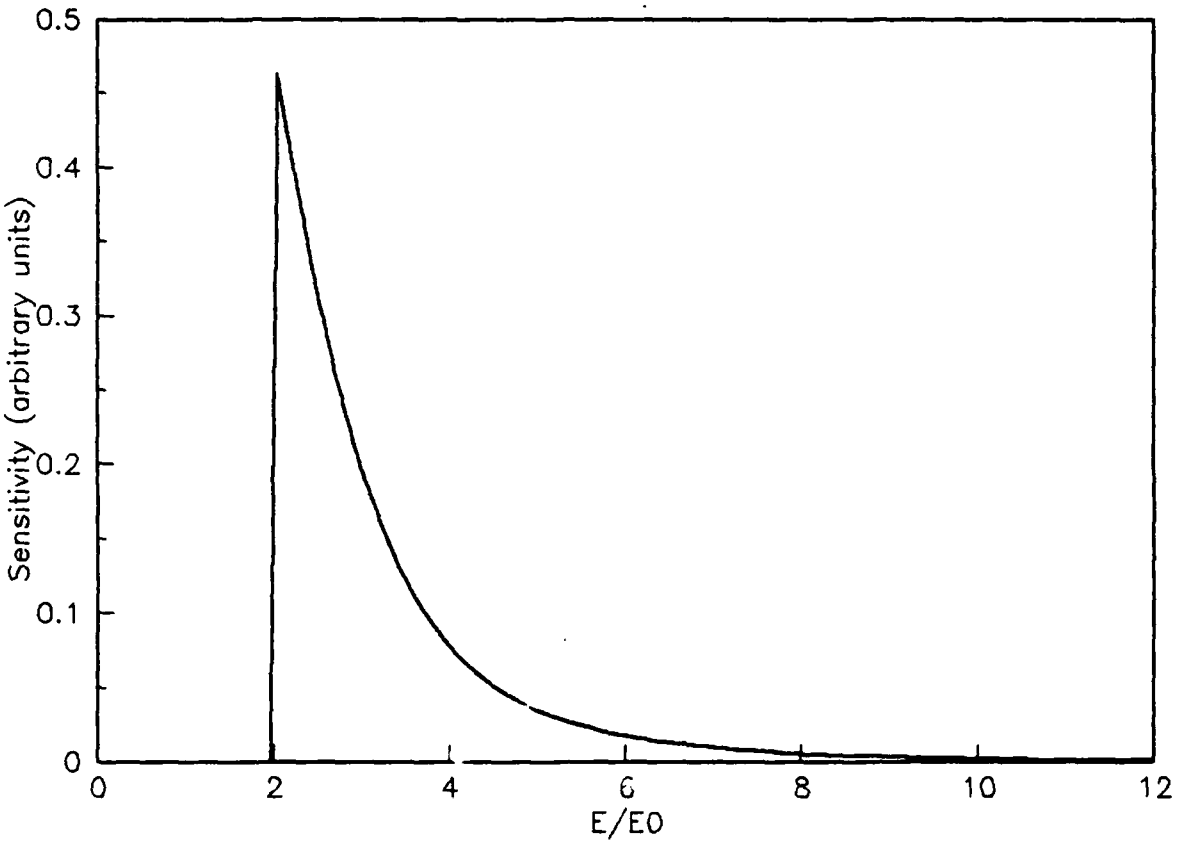


Figure 2: Typical Open Response Function

Since the photons are less energetic, they are typically below the fluorescer k-edge and have little probability of causing additional fluorescence. Photons of energy less than the filter k-edge and photons that survive the filter without reaction, react at the fluorescer in manner similar to those of the fluorescer detection system. Since the k-edges of the fluorescer and filter are different, there exists a an optimum energy for interaction with the system. Consequently, the response function of the "closed" system contrasts in shape to the "open" system and more closely approximates the

ideal or envelope shape. The detector envelope, commonly referred to as the "inband" response, occurs between the k-edges of the filter and the fluorescer. This envelope contains the majority of the response for the instrument. Again, varying the thickness and types of materials used for the filter and fluorescer changes the sensitivity of the system. Figure 3 is sketch of a typical filter fluorescer system. The functional form of the closed detection system response function used here is

$$R_i^{closed} = \begin{cases} 0 & E < E_i^0 \\ \left(\frac{1}{E}\right)\left\{1-\exp\left[-2\left(\frac{E_i^0}{E}\right)^3\right]\right\}\exp\left[-0.25\left(\frac{E_i^1}{E}\right)^3\right] & E_i^0 \le E < E_i^1 \\ \left(\frac{1}{E}\right)\left\{1-\exp\left[-2\left(\frac{E_i^0}{E}\right)^3\right]\right\}\exp\left[-1.5\left(\frac{E_i^1}{E}\right)^3\right] & E > E_i^1 \end{cases} \quad (4)$$

where

R_i^{closed} = the closed function of the i^{th} detector

E_i^0 = the k-edge of the fluorescer for the i^{th} detector

E_i^1 = the k-edge of the filter for the i^{th} detector

E = Energy

Figure 4 is an example of a typical closed response function. In figure 4, the detector k-edges occur at 2 E_0^1 and 4 E_0^1.

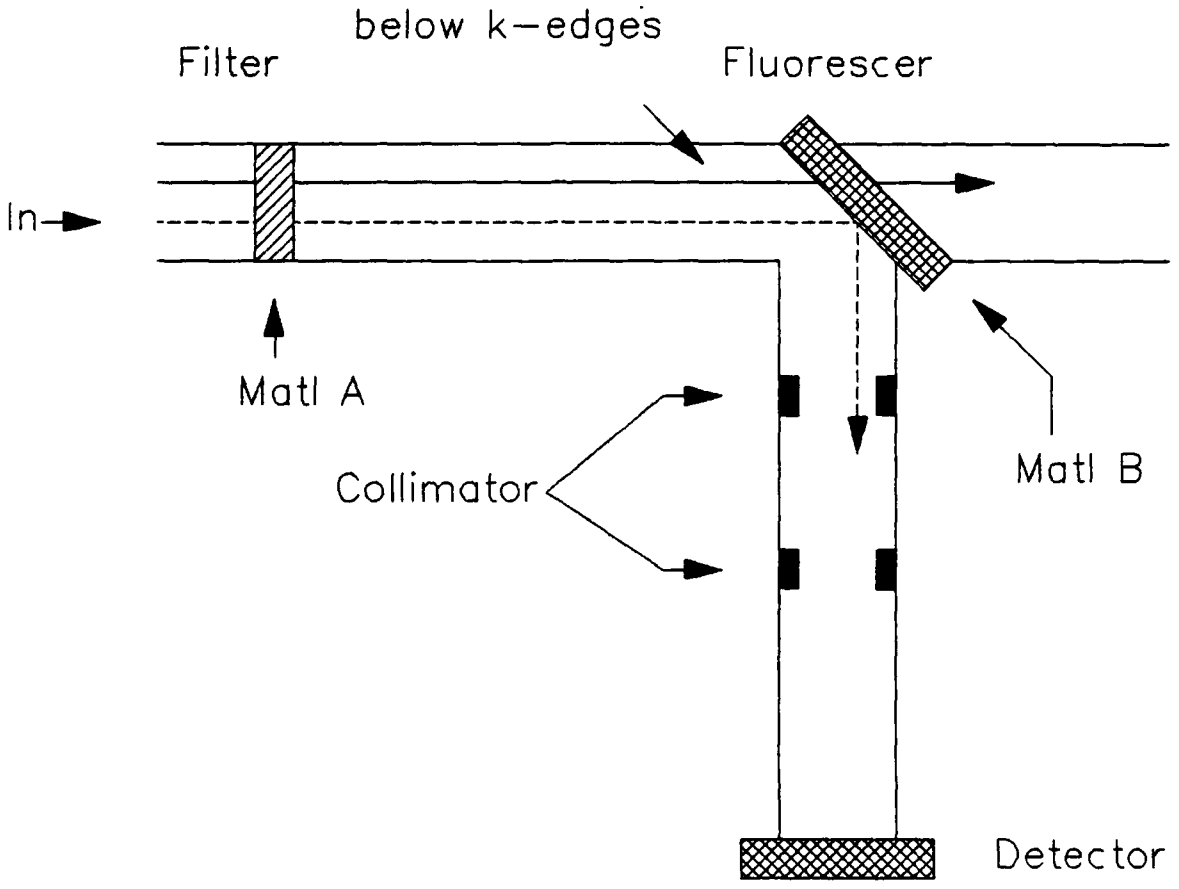


Figure 3: Typical Filter-Fluorescer Detection System

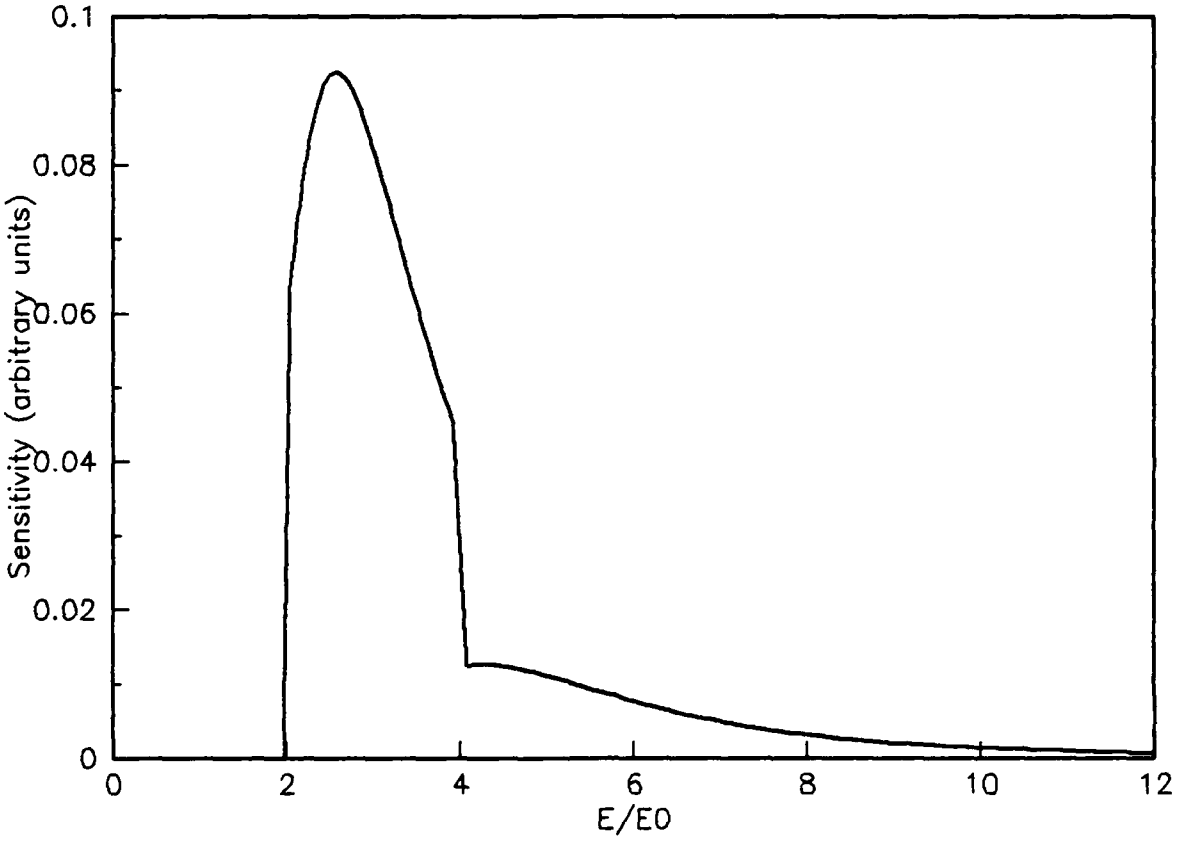


Figure 4: Typical Closed Response Function

Experiment Design

Consider a typical experiment which consists of an array of both open and closed detection channels chosen to provide overlapping responses. As stated earlier, seventeen detector channels were used for this study: eleven closed and six open channels. These were assumed to have a resolving power of about 1.5. The k-edges of the detectors are listed in Table 1.

Table 1: Detector K-Edges[1]

Det #	E_i^0	E_i^1	Det #	E_i^0	E_i^1
1	1	2	10	$16\sqrt{2}$	$32\sqrt{2}$
2	$\sqrt{2}$	$2\sqrt{2}$	11	32	64
3	2	4	12	1	
4	$2\sqrt{2}$	$4\sqrt{2}$	13	2	
5	4	8	14	4	
6	$4\sqrt{2}$	$8\sqrt{2}$	15	8	
7	8	16	16	16	
8	$8\sqrt{2}$	$16\sqrt{2}$	17	32	
9	16	32			

[1] E_i^0 and E_i^1 are ratios computed with respect to E_1^0.

The overlap of response functions and the inband response for the closed detection systems are seen clearly in Figure 5. Figure 6 shows the response functions for the collection of open detection systems.

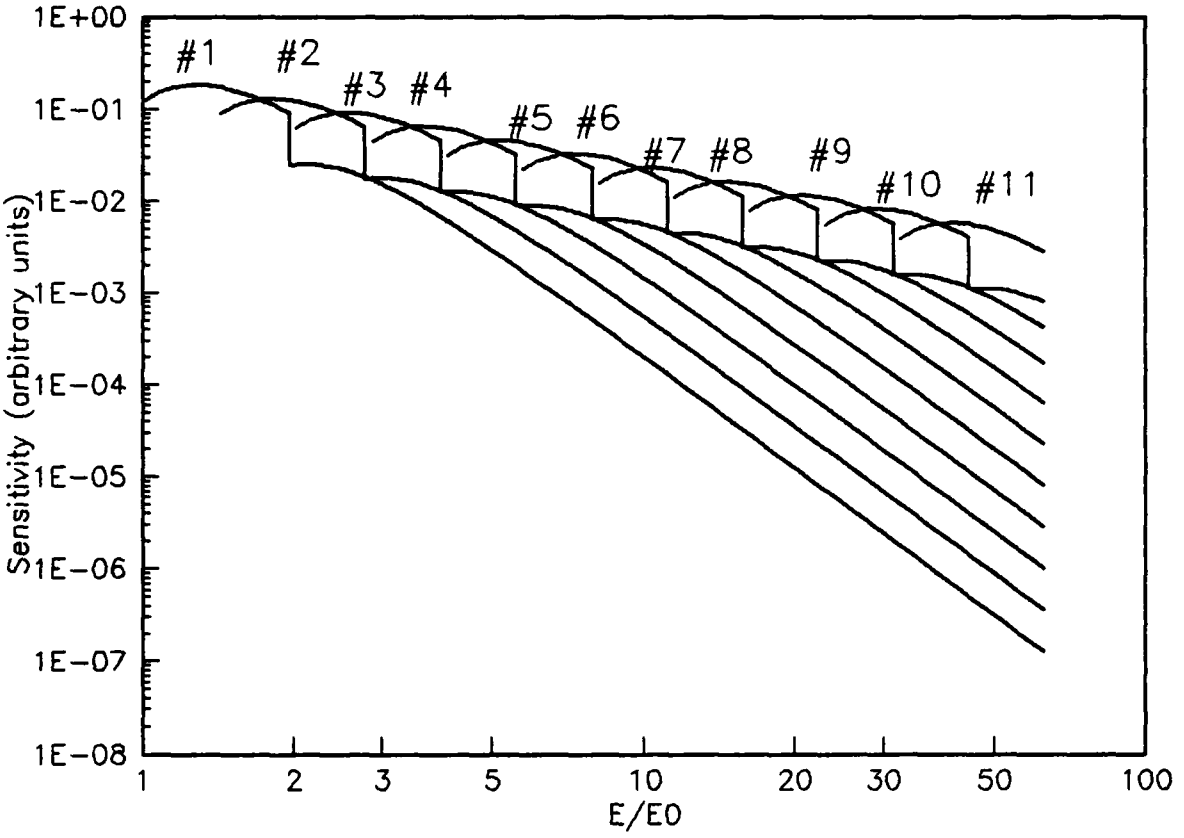


Figure 5: Closed Response Functions, Instruments 1-11

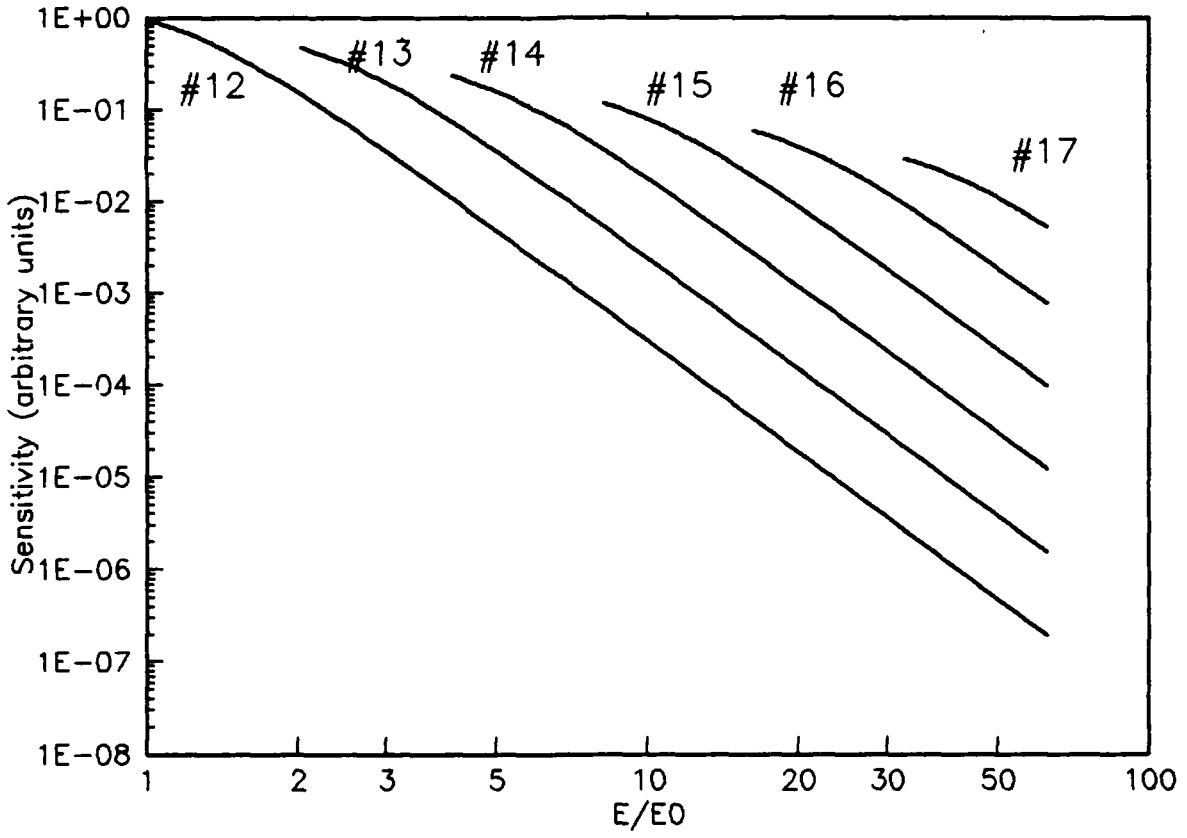


Figure 6: Open Response Functions, Instruments 12-17

Definition of Spectra and Signals

Spectra. For the purposes of this study, three spectra are required. They are the actual or exact spectrum, $S_s(E)$, the predicted spectrum, $S_p(E)$, and the unfolded spectrum, $S_U(E)$.

The exact spectrum, $S_s(E)$, is the actual spectrum incident on each detector. In a real application, this spectrum is obviously unknown and can never be known with complete certainty. The goal of the measurement and deconvolution process is to approximate $S_s(E)$ with $S_U(E)$.

The predicted spectrum $S_p(E)$, is the analyst's a priori estimation of $S_E(E)$. $S_p(E)$ is based on physics, calculations, and previous measurement of similar sources.

The unfolded spectrum $S_U(E)$, is the output of the deconvolution process. It is the unfolded approximation to the exact spectrum $S_E(E)$.

Signals. A signal is defined to be the measurable output of the detector system, such as a current or voltage pulse resulting from exposure to a spectrum. A signal is equal to the product of the response function and a spectrum, integrated over all energies. Four different types of signals arise from the convolution of the three previously defined spectra. These signals are the ideal signal, Y_i^{ideal}, the measured signal, Y_i^M, the predicted signal, Y_i^P, and the unfolded signal, Y_i^U.

The ideal signal, Y_i^{ideal}, is the signal that would be produced by the exact spectrum folded with an ideal instrument. The ideal signal (one with no measurement error or "noise") cannot be known since the exact spectrum is unknown and ideal response data can only be approximated by calibrated response data. In any case, the ideal signal, Y_i^{ideal}, is given by

$$Y_i^{ideal} = \int_0^\infty R_i^{ideal}(E) S_E(E) dE \tag{5}$$

where

$S_E(E)$- the exact spectrum

$R_i^{ideal}(E)$- the ideal (no measurement error) response function of the i^{th} detector

Y_i^{ideal}- the ideal signal of the i^{th} detector

The predicted signal, Y_i^P, is the signal expected based on the predicted spectrum. The predicted signal is given by

$$Y_i^P = \int_0^\infty R_i(E) S_P(E) dE \tag{6}$$

where

$S_P(E)$- the predicted spectrum

$R_i(E)$- the calibrated response function of the i^{th} detector

Y_i^P- the predicted signal of the i^{th} detector

The measured signal, Y_i^M, is the signal received including all the inherent uncertainties such as calibration and measurement error. The measured signal is given by

$$Y_i^M = \int_0^\infty \hat{R}_i(E) S_E(E) dE \tag{7}$$

where

$S_E(E)$- the exact spectrum

$\hat{R}_i(E)$- the actual response function of the i^{th} detector

Y_i^M- the measured signal of the i^{th} detector

The final signal is the unfolded signal, Y_i^U. The unfolded signal is a result of folding the unfolded spectrum with response data. The unfolded signal is given by

$$Y_i^U = \int_0^\infty R_i(E) S_U(E) dE \tag{8}$$

where

$S_U(E)$- the unfolded spectrum

$R_i(E)$- the calibrated response function of the i^{th} detector

Y_i^*- the unfolded signal of the i^{th} detector

The data that the analyst commonly receives from under ground nuclear effects simulations are not the signals themselves, but ratios of the signals. The two common ratios used are the measured-to-predicted ratio, b_i, which is given by

$$b_i = \frac{Y_i^M}{Y_i^P} \tag{9}$$

and the unfolded-to-predicted ratio, c_i, which is given by

$$c_i = \frac{Y_i^U}{Y_i^P} \tag{10}$$

Scaling results to predictions in this way removes details of the actual measurements. The actual picoamps or microvolts of signal are of no importance to the analyst, whereas the ratio to prediction is essential.

Sources of Error

Errors of Measurement. During the measurement process, two main sources of error exist. They are measurement error and calibration error. Measurement error consists of the uncertainty in the measured output of the detector. This is the cumulation of all the uncertainties related to the data collection process.

Calibration error is different from measurement error. It is the inherent uncertainty in the detector response function. Detector response may vary with factors such as time, temperature, etc. As a result, calibrated response data is used as a substitute to approximate the ideal response function. A calibrated response function is determined by exposing the detection system to known sources of x-rays, and is subject to the same types of measurement error discussed above.

Errors of Unfold. During the unfold process, two main sources for error exist. First, due to ill-posed nature of the integral equation, there are an infinite number of possible solutions; second, the approximation of the integral equation by a summation of discrete values leads to approximation error.

The Iterative Unfold Method

The iterative unfold process is a numerical technique for approximating a spectrum by deconvolution. This method starts by folding a trial guess at the spectrum (usually the predicted spectrum) with response data to generate unfolded signals. These unfolded signals are then compared to the experimentally measured signals. The trial spectrum is then altered by a two step process. First, the spectrum is modified through a fit process to force unfolded signals to move towards the experimentally measured signals. Second, the unfolded (fitted) spectrum is then smoothed through an averaging process to remove undesirable artifacts of the fit procedure. This two step process is then repeated as needed, each time using the output spectrum of the previous iterative step as the trial spectrum for the next step. Details of the fitting and smoothing processes are discussed next.

The "fit" procedure is an algorithm to force the unfolded signal over an energy interval to match the measured signal. This is accomplished effectively by scaling the guessed spectrum by the product of the measured-to-unfolded signals ratio and an appropriate response weighting function. Or more simply stated, the fit algorithm is basically a multiplication of the predicted result by a ratio of the measured-to--predicted result. Since the detection system responses overlap, the algorithm becomes a response weighted sum of the

product of the unfolded spectrum and the measured-to-unfolded ratio of signals. The "fit" algorithm as derived by Carter (2:20) is given by

$$S_U^{\left(n+\frac{1}{2}\right)}(E)dE = \frac{\left[\sum_{i=1}^{i_{max}} \beta_i \tilde{R}_i(E)\right]}{\left[\sum_{i=1}^{i_{max}} \tilde{R}_i(E)\right]} S_U^{(n)}(E)dE \tag{11}$$

where

$$\beta_i = \frac{Y_i^M}{Y_i^U} = \frac{Y_i^M/Y_i^P}{Y_i^U/Y_i^P} = \frac{b_i}{c_i} \approx \frac{\int_0^\infty \hat{R}_i(E) S_E(E) dE}{\int_0^\infty R_i(E) S_U(E) dE} \tag{12}$$

$$\tilde{R}_i(E) = \frac{R_i(E)}{Y_i^P} \tag{13}$$

and

$S_U^{\left(n+\frac{1}{2}\right)}$ is the unfolded spectrum at iteration n after fit, prior to smooth

$S_U^{(n)}$ is the unfolded spectrum at iteration n

$\tilde{R}_i(E)$ is the calibrated response function of the i^{th} detector

Equation 11 can be written more generally as,

$$S_U^{\left(n+\frac{1}{2}\right)}(E)dE = \sum_{i=1}^{i_{max}} \beta_i \left(\frac{\tilde{R}_i(E)}{\sum_{j=1}^{i_{max}} \tilde{R}_j(E)} \right) S_U^{(n)}(E)dE \qquad (14)$$

Or,

$$S_U^{\left(n+\frac{1}{2}\right)}(E)dE = \sum_{i=1}^{i_{max}} \beta_i w_i(E) S_U^{(n)}(E)dE \qquad (15)$$

where

$w_i(E)$= the fractional weight of the i^{th} instrument at energy E

The normalization of response functions to the predicted signals in Equation (14) was computationally convenient. However, it also allowed instruments with small predicted signals to have large input into the fitting process. Under certain conditions, the voice of the small signal instruments could be so large that the fitted signals would stop converging to the measured signals. As a result, the Carter fit algorithm was modified. The modification consisted of division of the numerator and denominator in Equation (14) by the unfolded-to-predicted ratio of signals. This change basically removed the prediction normalization and renormalized the response functions to the unfolded signals. This

modification provided a more rapid convergence to the measured signals and removed the scaling problems associated with poor predictions. The modified response weighting function is

$$w_i(E) = \frac{\tilde{R}_i(E)}{\sum_{j=1}^{i_{max}} \tilde{R}_j(E)} \tag{16}$$

where the response functions have been renormalized and are given by

$$\tilde{R}_i(E) = \frac{R_i(E)}{Y_i^U} \tag{17}$$

Due to abrupt changes in the detector response functions around the k-edges of the filter and fluorescer, large variations in scaling occur between adjacent detectors. These variations lead to discontinuities in the "fit" spectrum. Consequently, a smoothing algorithm is generally invoked to smooth out undesirable artifacts of the "fit" operation.

Since spectral discontinuities at detector k-edges are not likely to be features of the incident spectrum, these artifacts are removed using a smoothing algorithm. The

smoothing algorithm works in the following manner. First an averaging kernel, f, is introduced. This kernel is folded with fitted spectrum in the following manner,

$$S_U^{(n+1)}(E)dE = \frac{\int_0^\infty f(E,E')S_U^{\left(n+\frac{1}{2}\right)}(E')dE'}{\int_0^\infty f(E,E')dE'}dE \qquad (18)$$

where

$S_{(n+1)}$ = the spectrum after smoothing

This operation produces a smoother spectrum, but may not be as close to $S_E(E)$ as the fitted spectrum.

Negative results from the unfolded spectrum are obviously meaningless. However, the response functions used in this study are positive functions over all energies, consequently the fit routine weighting function, $w_i(E)$, is always positive. As a result, all fitted spectra obtained from the fit routine are non-negative. Similarly, the smoother weighting function, $f(E,E')$ is also non-negative. Therefore, all non-negative spectra obtained from the fit routine will remain non-negative through the smoothing process. Since neither the fit routine nor the smooth routine introduces any negative spectral artifacts, all non-negative input spectra will remain non-negative through the iterative unfold process.

In general, the fitting and smoothing operations are in competition. The fitting operation tends to drive the unfolded spectrum closer towards a spectrum that would produce measured signals when folded with response data, while the smoother averages out k-edge discontinuities and moves the unfolded spectrum towards a straight line. Using a symmetric averaging kernel over narrow energy ranges, the fitting operation can be made stronger than the smoothing operation. Consequently, repeated iterations through the iterative unfolder will produce an unfolded spectrum that generates signals that are increasingly closer to the measured signals. Since measurement errors are present in nearly every experiment and are an artifact of the experiment and not the incident spectrum, the measured signals are not ideal. As a result, the converging unfolded spectrum is not converging to the exact spectrum. Therefore, repeated iterations will at some point stop modeling the exact spectrum and begin modeling a noisy spectrum. In other words, the unfolded spectrum will become a better approximation for the exact spectrum with repeated iterations only to a point, then the uncertainties associated with the noise will tend to decrease the accuracy of the unfolded spectrum as an approximation. Without the exact spectrum for comparison, the point to stop modifying the unfolded spectrum is unclear.

In practice, the performance, or "goodness of fit", of the unfolding operation is measured by a χ^2 test statistic (2:28). The statistic is given by

$$\chi^2 = \sum_{i=1}^{instruments} \left(\frac{c_i - b_i}{\sigma_i} \right)^2 \qquad (19)$$

where

σ_i - the standard deviation of the measured-to-predicted ratio of the i^{th} instrument

Since the unfolded signals are converging to the measured signals, this statistic is a measure of the "goodness of fit" to a nonexistent noisy spectrum and is a function of the fit algorithm. Formulation of this statistic implicitly assumes normally distributed random independent variables and negligible systematic error. For the duration of this study the χ^2 test statistic formulated using measured signals (noise included) will be referred to as χ^2, while χ_i^2 will refer to the test statistic that uses the ideal (unknown) signals.

Since the iterative unfold process is continually driving the unfolded spectrum towards a noisy spectrum producing the measured signals, the χ^2 test statistic is monotonically decreasing with increasing iterations. Therefore, χ^2 provides only a measure of the fit to this hypothetical noisy spectrum and not the independent measure of the fit to the exact spectrum, S_i, which is needed. As a result of the dependent nature of χ^2, an almost arbitrary choice for the stopping point of the iterative unfold is made based on χ^2, observation of the unfolded spectrum, and user experience. The stopping point is usually chosen as

$$\chi^2 \approx \nu \tag{20}$$

where

ν the number of degrees of freedom and is usually taken to be on the order of the number of instruments.

The number of degrees of freedom, in general, is unclear. After the first iteration the data have been used to form an unfolded spectrum. Since the same data were used in forming the predicted spectrum as are used in the fitted spectrum, χ^2 is not independent. In fact, more degrees of freedom are adjusted than data are available. As a result, χ^2 has no degrees of freedom.

In the implementation of the iterative method, as accomplished by Carter (2), he states, "At this point, a judgement would have to be made to evaluate if the first iteration is reasonable." To date, what was reasonable had to be based on the experience of the user, and consequently, highly user dependent. The cross-validatory approach provides an objective method to estimate parameter optima, and therefore, provides a more independent test statistic.

Cross Validation

As stated earlier, the iterative method for spectral unfolding uses a test statistic, χ^2, to measure the "goodness of fit" of the unfolded spectrum. Since the exact spectrum is unknown, the measured-to-predicted ratio, b_i, is used in the χ^2 formulation. Each iteration of the iterative method

produces an unfolded spectrum that when folded with the appropriate response functions is closer to the measured signals. When noise is introduced the measured signals are non-ideal. Therefore, further iterations beyond some optimum will continue to decrease χ^2, but will likely be a less valid approximation to the exact spectrum. Consequently, χ^2 alone does not provide an independent stopping point for the iterative unfold process. The user would have to exercise "best judgement." This user dependent approach tends to lead to overfitting of the data by allowing χ^2 to fall to very low values. Cross-validation provides a method for removing the user dependencies from the process.

Cross-validation, sometimes referred to as predictive sample reuse, is basically a synthesis of the two well established processes of statistical cross-validatory assessment and function fitting (3:112, 4:320). Cross-validation consists of careful partitioning of the test data into sample subsets. These subsets are used in the formulation of a cross-validatory loss function. This loss function is minimized by varying the desired parameter over a wide range. The value of the parameter where the loss statistic is minimized is chosen as the estimate for the optimum parameter value. The following is a theoretical development of the cross-validatory approach.

Consider a collection of measurements, Y_i^M, which result from the use of n_i detection systems. Next, a means for predicting these signals is needed, sometimes referred to as a

prescription. A prescription need only provide a prediction for Y_i^M and might be quite general. In this case the prescription is the unfold process. Let the unfolded signal, Y_i^U, be the prediction for the measured signal, Y_i^M. In general, Y_i^U is a function of the detection system response, the unfolded spectrum, and other various parameters. A prescription for the unfolded signal would take the following form,

$$\{Y_i^U(\{R_j, b_j\}, S_P(E), S^0(E), \alpha)\} \tag{21}$$

where

R_j- the response function of the j^{th} detector

b_j- the measured-to-predicted ratio of the j^{th} instrument

$S_P(E)$- the predicted spectrum

$S^0(E)$- the initially guessed spectrum

α- the vector of parameters used in the unfold model

j is in the subset of instruments with the i^{th} instrument omitted

The traditional, or naive, measure of the ability to predict the exact spectrum incident on the detectors is the following,

$$\bar{L}(\alpha) = \sum_{i=1}^{n_i} \left(\frac{Y_i^M - Y_i^U(\{R_j, b_j\}, S_P(E), S^0(E), \alpha))}{\sigma_i Y_i^P} \right)^2 \tag{22}$$

where

σ_i- the uncertainty in the measure-to-predicted ratio

simplifying,

$$\bar{L}(\alpha) = \sum_{i=1}^{n_i} \left(\frac{b_i - c_i(\{R_j, b_j\}, S_P(E), S^0(E), \alpha)}{\sigma_i} \right)^2 = \chi^2(\alpha) \tag{23}$$

Cross-validation prescribes partitioning the data set into n_i subsets of $n_i - 1$ signals. As a result, seventeen data subsets are created containing sixteen signals. Each subset of data is used to formulate a prediction for the omitted data. Consequently, the cross-validatory choice of α (in this case iteration number) is the value α^* that minimizes the following

$$Q_i(\alpha) = \sum_{i=1}^{n_i} \left(\frac{Y_i^M - Y_i^U(\{R_j, b_j\}, S_P(E), S^0(E), \alpha)}{Y_i^P \sigma_i} \right)^2 \tag{24}$$

simplifying,

$$Q_i(\alpha) = \sum_{i=1}^{n_i} \left(\frac{b_i - c_i(\{R_j, b_j\}, S_P(E), S^0(E), \alpha)}{\sigma_i} \right)^2 \tag{25}$$

where

j is in the subset of instrument with the i^{th} instrument omitted

The calculation of the unfolded-to-predicted ratio, c_i, used in the formulation of the χ^2 test statistic uses the measured data. As a result, c_i is a function of the measurements, and thus, a function of the measured-to-predicted ratio, b_i. Therefore, χ^2 gives an independent measure only at iteration zero (no data used) when the predicted spectrum is used as the guess. All further iterations are dependent measures and the remaining degrees of freedom are unclear. In contrast, Q_i is formulated using a subset of the measured data. The omitted data are then predicted using the subset. In this case c_i is not a function of b_i, since a new c_i is formulated from the unfolded spectrum with the original c_i omitted. Consequently, Q_i is a more independent statistic. Minimization of Q_i provides the cross-validatory prediction for the optimum stopping point of the unfold method.

III. Computer Program Implementation

Introduction

Section II discussed the theoretical development of the iterative method for deconvolution of spectra, as well as, the methodology involved in cross-validation. In this section, this methodology will be implemented into a numerical model. The model will consist of input of data, noise simulation, iterative unfolding, and cross-validation. Each of these topics is discussed independently.

Input Data

The data available to the analysts at the conclusion of weapons effects simulation are the following: the predicted spectrum, $S_p(E)$, the calibrated response functions, $R_i(E)$, the measured-to-predicted ratios, b_i, and the estimated standard deviation of the measured-to-predicted ratios, σ_i. For the purposes of this study, these quantities were simulated.

In order to generate the necessary data for this study, an exact spectrum, $S_e(E)$, was simulated. This spectrum was then folded with the appropriate response functions to generate measured signals. This technique was repeated for the predicted spectrum. Using both the measured signals and the predicted signals, measured-to-predicted ratios were constructed.

The calibrated response functions, $R_i(E)$, were assumed to be exact. Errors in calibration are indistinguishable from

errors in measurement, and are accounted for by the addition of random noise to the measured-to-predicted ratios discussed above. Noise simulation is discussed in the next section.

The fractional standard deviations of detection systems were assumed constant at 0.15 to provide correlation to previous work by Daniels (1:3-1) and Carter (2:32). A thorough discussion of sources of error is covered in section II.

The predicted spectrum is data given to the analyst along with test data. Presumably, it is close to the exact spectrum. Consequently, the predicted spectra used in the study are always chosen close to the corresponding actual spectra, and are used as the first guess for the iterative unfold method.

Error Considerations

Unfold Errors. To reduce the errors in approximating integrals with summations, a geometric energy bin spacing scheme was used in lieu of linear spacing. The purpose was to reduced computational costs. In addition, the geometrically spaced energies allowed higher resolution of the lower k-edge detection systems. Using a 120 bin geometrically spaced grid, the relative error in approximating the needed integrals was less than 10^{-3}.

Noise Simulation. For the purposes of this study all measurement errors were simulated. These errors were assumed to be normally distributed and produced by randomly sampling a Gaussian probability distribution function. This sampling

was approximated by the sum of continuous uniform distributions. As a result, the noise added to the measure-to-predicted ratio is given by (2-15),

$$Z_i = \sigma_i \left(\sum_{n=1}^{6} x_n - 6 \right) \tag{26}$$

where

Z - the noise added to measure-to-predicted ratio

x_n - a pseudo-randomly generated number on [0,1)

σ_i - the standard deviation of the measured-to-predicted ratio

Iterative Unfold Method

Once the necessary information has been simulated, an algorithm for unfolding spectra was needed. The unfold process used for this study was the iterative unfold method. As stated in Section II, an iterative unfold is basically a two step process. The first step is to modify the trial or guessed spectrum through a "fit" procedure. Next the fitted spectrum is "smoothed" through an averaging process to remove discontinuities resulting from the fit procedure. The unfolded spectral output is then used as an approximation to the exact spectrum. The process is then repeated using each resulting unfolded spectrum as the trial spectrum for next unfold.

The first step in the iterative unfold process is the fit

process. The fit process consisted of summing the product of an instrument weighting function with the corresponding measured-to-unfolded ratio and is given by

$$S_U^{\left(n+\frac{1}{2}\right)}(E)dE = \sum_{i=1}^{i_{max}} \beta_i w_i(E) S_U^{(n)}(E)dE \tag{27}$$

In this study two different weighting functions were tested. They were the fractional response function approach used by Carter (2), which is given by

$$w_i(E) = \frac{\tilde{R}_i(E)}{\sum_{j=1}^{i_{max}} \tilde{R}_j(E)} \tag{28}$$

and the modified fractional response function approach which is given by

$$w'_i(E) = \frac{\tilde{R}_i(E)c_i}{\sum_{j=1}^{i_{max}} [\tilde{R}_j(E)c_j]} \tag{29}$$

The major difference between the two approaches is that the modified approach removes the prediction normalization of the response functions and renormalizes them to the unfolded signals. The modified approach revises the weights of each instrument after each iteration, taking advantage of the

previous iteration's unfolded spectrum which is usually a better approximation to the exact spectrum. The response weighting function given in Equation (26) was chosen for this study because it gave more rapid fit to the data and reduced computation time. Furthermore, it reduced the prediction scaling problems discussed in Section II.

The second step in the iterative unfold process is the smoothing process. For this study a three-bin averaging approach was used. The three-bin averaging approach gave equal weighting to the central and two adjacent energy-bins on a three bin interval. The spectral values were averaged over the three-bin interval and used as the value for the center energy bin. The averaging method employed compensates for truncation at the ends of the energy interval (1 E_0^j to 64 E_0^j) in a conservative manner by weighting the energy-bin at the truncation twice the adjacent energy bin. A weighted average is then performed on the two energy-bins to produce the smoothed value for the bin at the truncation.

Two energy bin spacing schemes were explored. They were linear spacing and geometric spacing. Geometric energy bin spacing gave a much more refined energy grid at low energies and consequently approximation errors were reduced significantly for the same number of energy bins. As a result, geometric spacing was used for the duration of this study.

Cross-Validation

Cross-validation was implemented to provide an independent means of stopping the iterative unfold process. Its goal was to remove the user-dependencies from the iterative unfold approach.

Cross-validation consisted of repeated application of the iterative unfold algorithm omitting each instrument, one at a time. The subset of signals were then used to form an unfolded spectrum. After the iterative unfolder returned an unfolded spectrum, the spectrum was then folded with the appropriate response function to form an unfolded signal. The unfolded signal was then used as the estimator for the omitted signal and a cross-validatory loss statistic was calculated.

CvIter.bas

CvIter.bas is the final version of the cross-validation/iterative unfolding code. The computer code is a Microsoft Quick Basic V4.5 code which cross-validatorily predicts the optimum stopping point for the unfold process. The code provides for linear or geometric energy bin spacing and three-bin averaging in the smoother. The following section contains a discussion of the results and validation of this program. The source code is contained in Appendix A. A pseudo code is contained in Appendix D.

IV. Results and Discussion

Introduction

In Section III the implementation of cross-validation coupled with the iterative unfold method was discussed. In this section the following topics will be presented: The validation cases, the test cases, the effects of noise, and the performance of the cross-validatory test statistic compared to other common test statistics.

Validation Cases

Validation cases were constructed to ensure the proper operation of the iterative unfold routine. The goal of the validation was to retrieve a close approximation to the exact spectrum from a guessed spectrum after repeated iterations. As a result, there was no simulated error added to the measured signals. In the absence of the error, the iterative unfold scheme should be able to reproduce a close approximation to the exact spectrum, and validate the proper operation of the iterative method.

In order to maintain continuity with previous work by Carter (2), similar validation cases were constructed. The three validation cases are discussed in detail below. Each case used a combination of Planckian functions given by (6:5),

$$P(E_0) = C_i \frac{15}{(\pi E_0)^4} \frac{E^3}{\exp(E/E_0) - 1} \qquad (30)$$

where

C_i = the scaling factor for Planckian i

Validation Case 1 (VC1). The purpose of this validation case was to show the convergence towards the exact spectrum after repeated iterations. In case VC1, the actual and guessed (predicted) spectra were both one temperature Planckians. The actual spectrum was a 3 E_0' Planckian with a scaling factor of 0.75. The guessed spectrum was a 4 E_0' Planckian with 1.0 scaling.

Validation Case 2 (VC2). In case VC2, the actual and guessed spectra were identical. The spectra were one temperature Planckians at a temperature of 3 E_0' and scaling of 1. This should demonstrate that modifications to the exact spectrum will change the spectrum after an unfold.

Validation Case 3 (VC3). Validation case VC3 uses a two temperature Planckian for the exact spectrum and uses a one temperature Planckian as the guess spectrum. The exact spectrum had temperatures at 1.2 E_0' and 6 E_0' with scaling 1.0 and 2.0, respectively. The guessed spectrum had a temperature of 3 E_0' and a scaling of 2.5.

Validation Results. Two variations were attempted in each validation case. First, the iterative unfold routine was implemented using only the fit algorithm. Next, the iterative unfold routine was implemented with fit and smooth algorithms in tandem. The results of the validation cases are discussed below.

For validation case 1 using only the fit algorithm, each successive iteration through the iterative unfolder produced an unfolded spectrum which reduced the error in fit to the exact spectrum. Although, the fit error was smaller in each successive fit, k-edge discontinuities developed after the first iteration and slowed convergence to the exact spectrum. Figure 7 shows the resultant unfolded spectra from validation case 1.

Next, the smooth routine was turned on and an iterative unfold of validation case 1 was repeated. Although the smooth routine competes with the fit, it tends to smooth out k-edge discontinuities returning a spectrum with less abrupt discontinuities at k-edge boundaries. Figure 8 shows the results of unfolding case VC1 using the fit and smooth routine together.

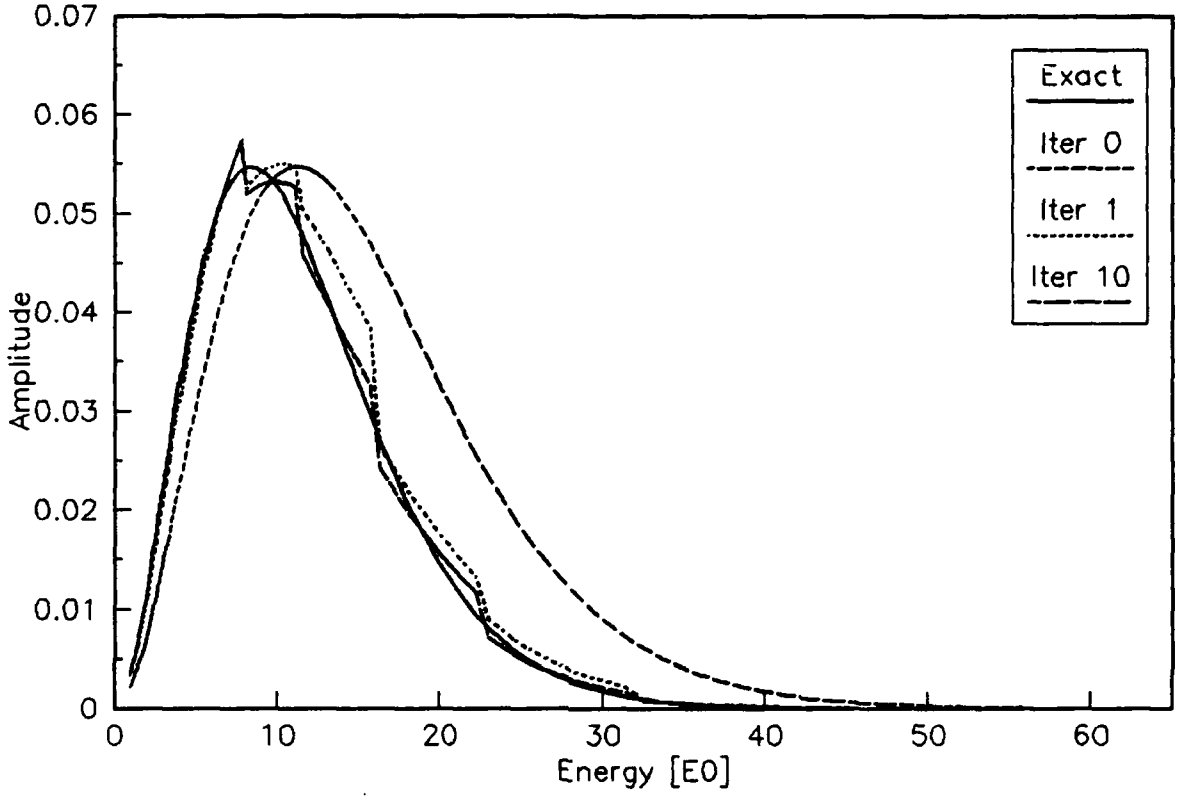


Figure 7: Case VC1 (smoother off)

Validation case 2 used the same exact and predicted spectra. In this case the iterative unfold routine using both the fit alone and the fit/smooth routines in tandem, produced spectra that were nearly indistinguishable from the exact spectrum. Consequently, the iterative unfold routine does not drastically distort a correct prediction.

In validation case 3, a two temperature Planckian was used as the exact spectrum, while a one temperature Planckian was used as the guess. Using the fit algorithm alone, the

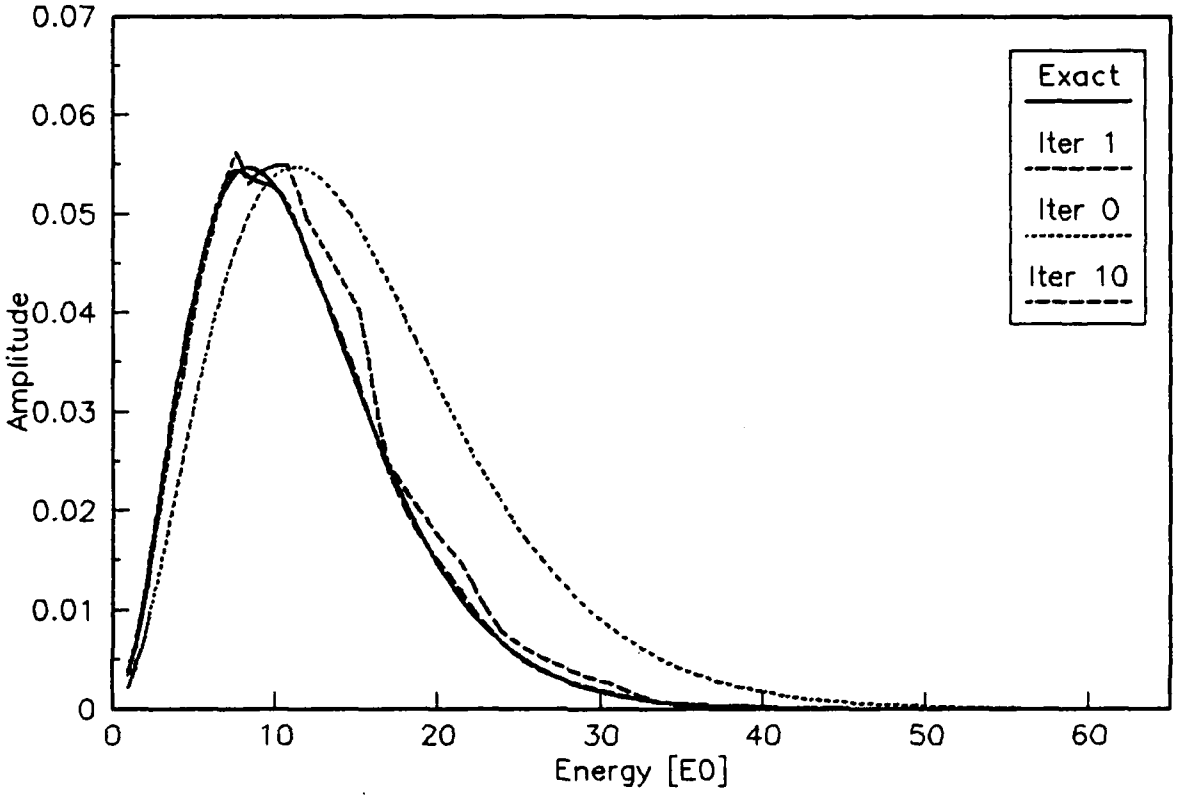


Figure 8: Case VC1 (smoother on)

iterative unfold routine produced spectra which fit closer to the exact spectrum with each iteration. But again, convergence was slowed by the introduction of k-edge discontinuities. In Figure 9, the unfolded spectra after the first and tenth iterations are pictured.

Using the fit and smooth routine together, case VC3 was repeated. Figure 10 shows the unfolded spectra at one, two, and ten iterations. Each successive iteration produces a spectrum that fits closer to the exact. Also, increasing

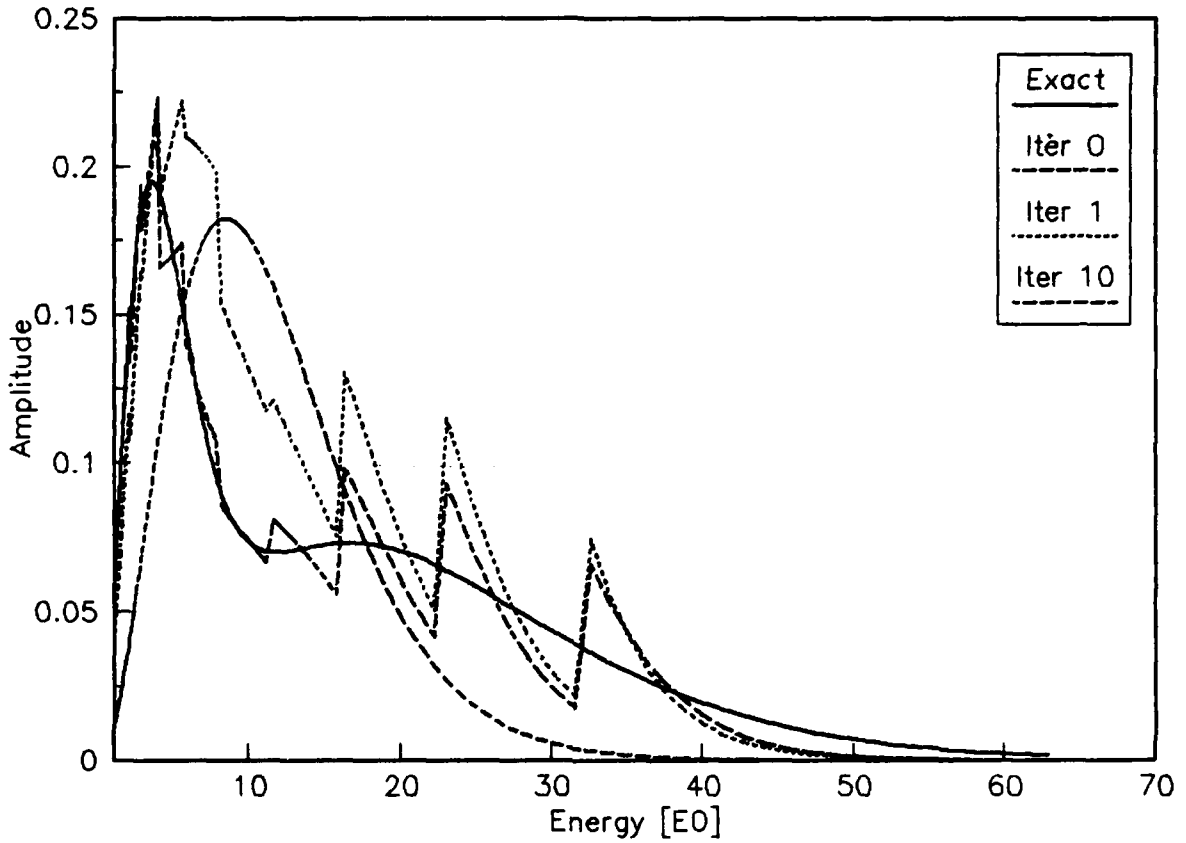


Figure 9: Case VC3 (smoother off)

iteration enhances the smoothing of the spectrum and lessens the impact of the k-edge discontinuities. In fact, after iteration 1 the smoothed spectrum gave a better fit than using the fit algorithm alone.

In each validation case, using either the fit alone or the fit/smooth routines together, the iterative unfold method returned an unfolded spectrum that better fit the exact spectrum with increasing iteration. The fit error, χ^2, for cases VC1 and VC3 are tabulated in Table 2. Table 2 shows that in

each case χ^2 is monotonically decreasing, giving an increasing fit. It was also seen that the iterative unfold routine would not significantly distort a prediction which corresponded to the exact spectrum.

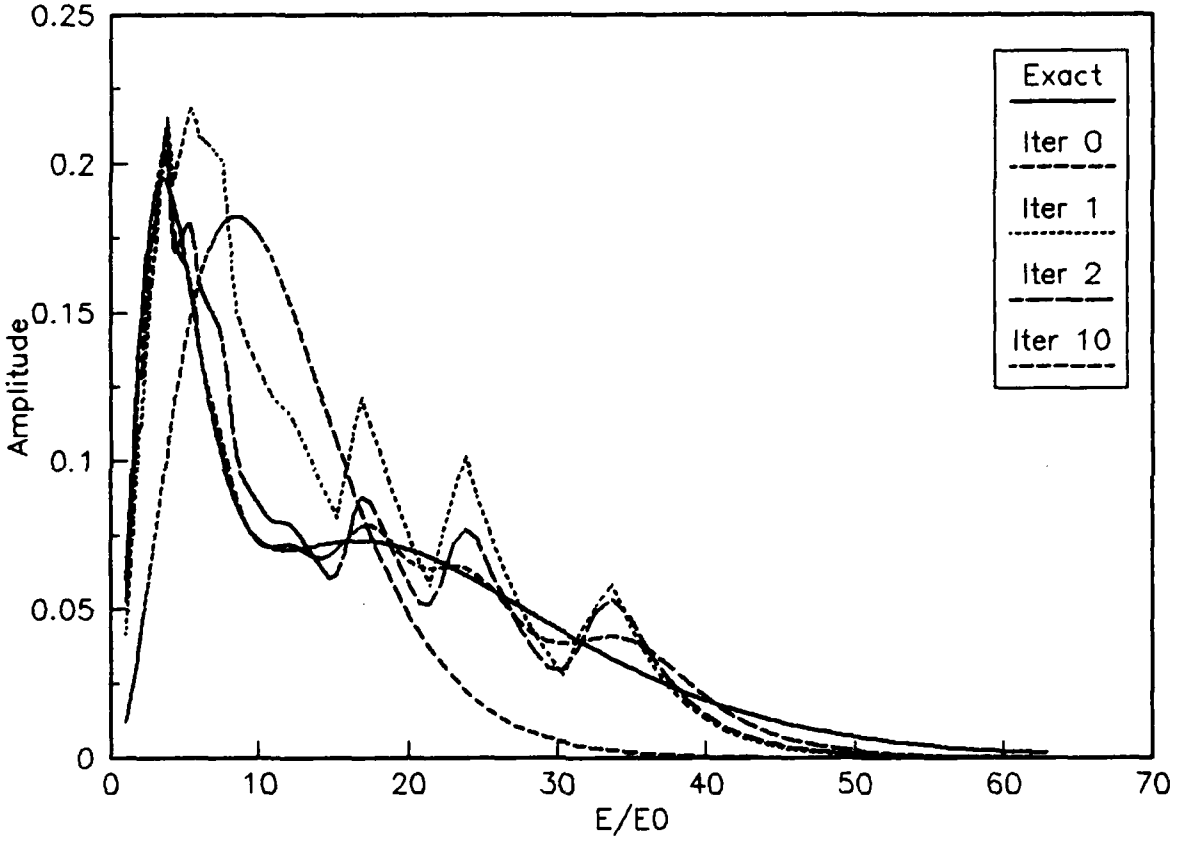


Figure 10: Case VC3 (smoother on)

Table 2: χ^2 for Cases VC1 and VC3

Iter Number	VC1 fit only	VC1 fit & smooth	VC3 fit only	VC3 fit & smooth
0	6057.600	6057.600	306.330	306.330
1	10.479	16.390	80.787	81.704
2	0.414	0.528	5.418	5.207
3	0.240	0.296	2.023	1.416
4	0.202	0.230	1.601	0.886
5	0.182	0.202	1.452	0.693
6	0.168	0.182	1.354	0.572
7	0.157	0.169	1.276	0.481
8	0.147	0.159	1.209	0.409
9	0.138	0.151	1.149	0.351
10	0.131	0.144	1.096	0.303

Cross-validation (CV)

As stated earlier, cross-validation partitions data into sample subsets to estimate model parameters. The optimization process centers around the minimization of the Q_t test statistic. In words, Q_t is a measure of the ability to predict the unfolded signal, Y_t'', with Y_t'' omitted from the sample set.

Validation of CV. In the absence of sources of noise such as measurement error, the optimum stopping points for the iterative unfold routine are clear. First, if the guessed spectrum is identical to the exact spectrum, then any fit/smooth modifications to the guessed spectrum will make the unfolded spectrum a poorer approximation to the exact spectrum. Consequently, cross-validation should predict no modification to the spectrum as the optimum, resulting in a local minimum to the Q_t function at zero iterations. Second, if the guessed spectrum is different than predicted spectrum,

then each successive iteration will produce a spectrum that is an improved fit to the measurements. As a result, the behavior of Q_i should be monotonically decreasing in nature exhibiting no local minima. Therefore, two validation cases were constructed.

Validation case 1 (VCV1) and validation case 2 (VCV2) were basically a repeat of iterative unfold validation cases VC2 and VC3, respectively. In this attempt cross-validation was used and the results were compared.

Results of CV Validation. In VCV1, the Q_i function exhibited a local minimum at iteration zero as did χ^2_f. The results are presented in Table 3.

Table 3. Case VCV1: Comparison of Q_i to χ^2_f

Iteration Number	χ^2_f fit/smooth	Q_i fit/smooth
0	0.000	0.00
1	0.008	2.14
2	0.009	2.98
3	0.009	3.64
4	0.009	4.37
5	0.010	5.16
6	0.010	5.99
7	0.010	6.86
8	0.010	7.77
9	0.010	8.71
10	0.010	9.66

In VCV2, Q_i behaved in a fashion similar to χ^2_f, decreasing monotonically with increased iteration number. The results of VCV2 are contained in Table 4. It is important to note

that, although, χ_f^2 is approaching zero, Q_i does not. This is because Q_i is sensitive to the fact that the unfold is fitting the measurements used in unfolding, but is not actually fitting the spectrum.

Table 4. Case VCV2: Comparison of Q_i to χ_f^2

Iteration Number	χ_f^2 fit/smooth	Q_i fit/smooth
0	306.33	306.30
1	81.71	129.90
2	5.21	19.64
3	1.42	9.48
4	0.89	6.89
5	0.69	5.72
6	0.57	4.97
7	0.48	4.41
8	0.41	3.95
9	0.35	3.56
10	0.30	3.22

In the absence of noise, the minimization of the cross-validatory test statistic, Q_u predicts the optimum stopping point for the iterative unfold routine in both validation cases.

Analysis Approach. As stated in Section II, the iterative unfold routine modifies the guessed spectrum. The modification consists of scaling the guessed spectrum in a manner that would produce unfolded signals that converge to the measured signals after many iterations. As a result, the behavior of an independent statistic χ_f^2 would exhibit a local minimum at the point when the modified signals are closest to

the exact signals. But a better measure of how well an unfolded spectrum approximates the exact spectrum are spectral norms (discussed below).

Although in practice χ_i^2 is unknown, for the subsequent test cases χ_i^2 (constructed so that it is known) will be used as an analysis tool for comparison to the Q_i function. In addition, two spectral norms will be calculated to determine the degree to which the unfolded spectrum compares with its exact counterpart. The first spectral norm, L_a, is a measure of absolute error in the spectrum and is given by

$$L_a = \frac{1}{n}\sum_{j=1}^{n}(S_j^U - S_j^E)^2 \tag{31}$$

where

S_j^E - the spectrum value in the jth energy bin for the exact spectrum

S_j^U - the spectrum value in the jth energy bin for the unfolded spectrum

n - the number of energy bins

The second spectral norm, L_r, is a measure of the relative error and is given by

$$L_r = \frac{1}{n}\sum_{j=1}^{n}\left(\frac{S_j^U - S_j^E}{S_j^E}\right)^2 \tag{32}$$

Two test cases were chosen. In each case, the fractional measurement uncertainty was assumed to be 0.15 for all instruments. In the first test case (TC1) the actual and guessed (predicted) spectra were both one temperature Planckians. The exact spectrum was a 3 E_0^1 Planckian with a scaling factor of 0.75, same as VC2, except with noise added to the signals. In test case 2 (TC2), a two temperature Planckian is used for the exact spectrum. The temperatures for the exact spectrum in TC2 were set at 1.2 E_0^1 and 6 E_0^1 with scaling 1.0 and 2.0, respectively, same as VC3, but with noise added.

In each case the analysis consisted of comparison of the absolute error in unfolded spectral approximation based on three iteration stoppage criteria. Theses criteria were values of χ^2 equal to the number of instruments (17), χ^2 equal to half the number the number of instruments (8.5), and minimization of Q_i. In these cases the guessed spectrum and simulated noise set were systematically varied to obtain unfolded spectra. In addition, the smoother was turned off in some cases to view the effects of the fit routine alone. Next, ten repetitions of test case 2 (smoother on) were performed varying the sets of noisy signals to observe the effects of random noise on the ability of the loss statistics to predict an iterative stopping point.

Results of Test Case 1 (TC1). In case TC1.0, the guessed spectrum was a one temperature Planckian with scaling of 1.0 and a temperature of 4 E_i^0. For case TC1.0, the iterative unfold routine used the fit algorithm alone. The results are

summarized in Table 5.

Table 5 shows that χ^2 monotonically decreases, as expected, with increasing iterations. χ^2_s, Q_u and L_a exhibit minimums at two, five, and one iterations, respectively. Figure 11 depicts the unfolded spectra at each of the above iteration numbers.

Table 5: Test Statistics for Case TC1.0 (smoother off)

Iter	χ^2	χ^2_s	Q_t	L_a	L_r
0	8.5340E+03	8.1772E+03	8.5340E+03	8.1536E-05	1.8260E+02
1	3.9155E+01	8.8238E+00	1.1020E+02	8.3206E-06	1.3558E+00
2	1.8417E+01	5.0957E+00	6.0925E+01	1.2446E-05	6.5781E-01
3	1.5840E+01	7.2719E+00	5.9419E+01	1.9508E-05	4.8490E-01
4	1.4190E+01	8.9996E+00	5.9146E+01	2.6439E-05	3.5812E-01
5	1.3009E+01	1.0369E+01	5.9117E+01	3.2692E-05	2.6557E-01
6	1.2112E+01	1.1459E+01	5.9243E+01	3.8233E-05	1.9902E-01
7	1.1397E+01	1.2334E+01	5.9464E+01	4.3140E-05	1.5197E-01
8	1.0804E+01	1.3047E+01	5.9740E+01	4.7509E-05	1.1947E-01
9	1.0297E+01	1.3635E+01	6.0047E+01	5.1427E-05	9.7861E-02
10	9.8527E+00	1.4127E+01	6.0371E+01	5.4972E-05	8.4372E-02

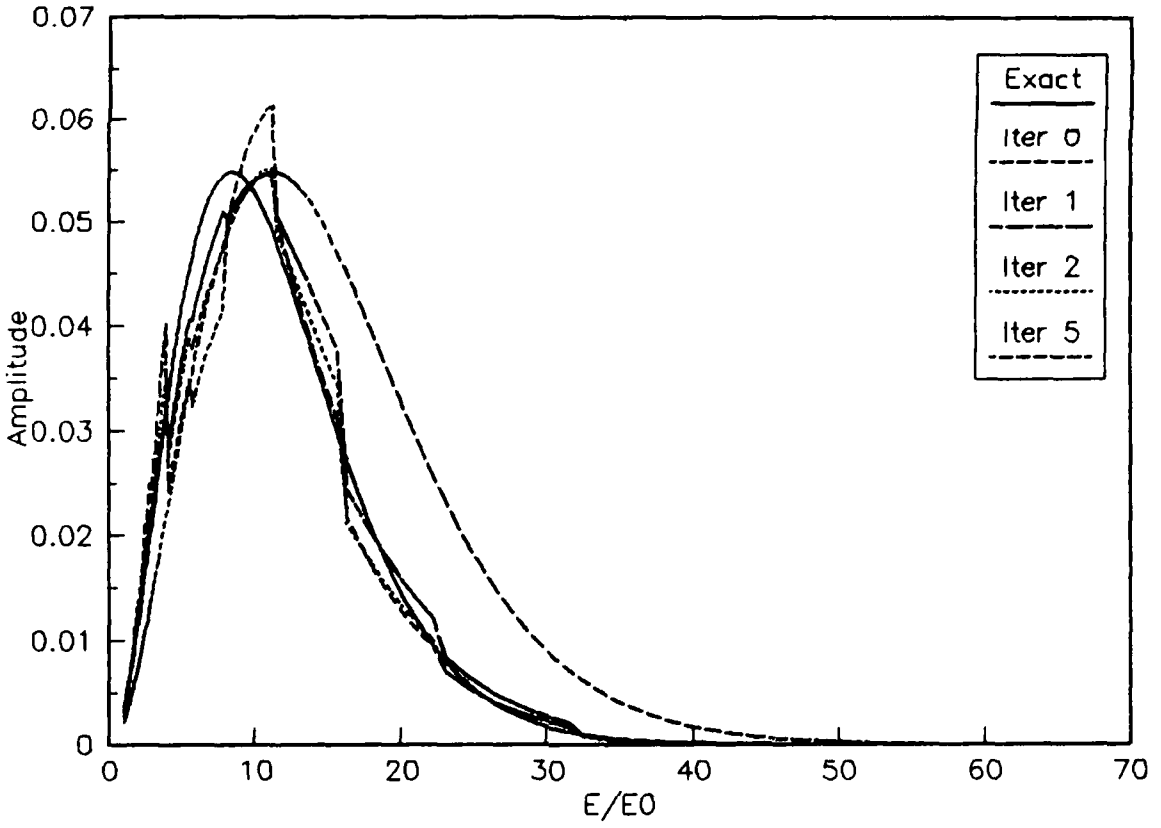


Figure 11: Case TC1.0 (smoother off)

Case TC1.1 consisted of modification to the guessed spectrum from case TC1.0. In this case, the guessed spectrum was identical to the exact spectrum at a temperature 3 E_i^0 and scaling of 0.75. Again, the iterative unfold routine used only the fit algorithm. The results are summarized in table 6.

Since the ideal signals are not equal to the measured signals, Table 6 shows a monotonically decreasing χ^2, as the ideal signals converge to the measurement, indicating proper

operation of the fit routine. Note that each of the test statistics tabulated, excluding χ^2, exhibits a minimum at zero iterations (the guessed spectrum).

Table 6: Test Statistics for Case TC1.1 (smoother off)

Iter	χ^2	χ^2_f	Q_l	L_a	L_r
0	3.0559E+01	0.0000E+00	3.0559E+01	0.0000E+00	0.0000E+00
1	1.8523E+01	3.2497E+00	4.2832E+01	5.6528E-06	6.0993E-03
2	1.4945E+01	5.6969E+00	4.3917E+01	1.3475E-05	1.4906E-02
3	1.2895E+01	7.8514E+00	4.3797E+01	2.1434E-05	2.5138E-02
4	1.1607E+01	9.6021E+00	4.3831E+01	2.8583E-05	3.5752E-02
5	1.0727E+01	1.0988E+01	4.4088E+01	3.4755E-05	4.6276E-02
6	1.0079E+01	1.2082E+01	4.4496E+01	4.0036E-05	5.6491E-02
7	9.5732E+00	1.2951E+01	4.4990E+01	4.4571E-05	6.6303E-02
8	9.1574E+00	1.3650E+01	4.5529E+01	4.8506E-05	7.5683E-02
9	8.8024E+00	1.4219E+01	4.6085E+01	5.1963E-05	8.4633E-02
10	8.4901E+00	1.4689E+01	4.6645E+01	5.5044E-05	9.3172E-02

In case TC1.0s, the exact and guessed spectra were identical to case TC1.0. This case is a duplicate of case TC1.0, except the smooth routine was included in the iterative unfold algorithm. The results are summarized in Table 7.

Table 7 shows that through ten iterations χ^2 has decreased from its original value of 30.56 to 8.49. In this case, χ^2_f, Q_l and L_a all exhibit minimums. These minimums occur at two, eleven, and one, respectively. Figure 12 shows the unfolded spectra at these iteration numbers. Note that smoothing provides a better fit to the measured signals than fitting alone (case TC1.0) at all iterations. In addition, the minimum χ^2_f is lower in the smoothed case.

Table 7: Test Statistics for Case TC1.0 (smoother on)

Iter	χ^2	χ^2_I	Q_l	L_a	L_r
0	8.5340E+03	8.1772E+03	8.5340E+03	8.1536E-05	1.8260E+02
1	4.8621E+01	1.4516E+01	1.2391E+02	7.7213E-06	1.6642E+00
2	1.9210E+01	4.5851E+00	6.0110E+01	1.1447E-05	7.8379E-01
3	1.6395E+01	6.6176E+00	5.6947E+01	1.8003E-05	6.1829E-01
4	1.4616E+01	8.2474E+00	5.5441E+01	2.4312E-05	4.9087E-01
5	1.3451E+01	9.5162E+00	5.4557E+01	2.9841E-05	3.9347E-01
6	1.2618E+01	1.0510E+01	5.3970E+01	3.4572E-05	3.1752E-01
7	1.2008E+01	1.1291E+01	5.3607E+01	3.8595E-05	2.5845E-01
8	1.1545E+01	1.1909E+01	5.3385E+01	4.2014E-05	2.1253E-01
9	1.1185E+01	1.2402E+01	5.3260E+01	4.4924E-05	1.7703E-01
10	1.0898E+01	1.2797E+01	5.3203E+01	4.7410E-05	1.4982E-01
11	1.0667E+01	1.3118E+01	5.3193E+01	4.9541E-05	1.2925E-01
12	1.0476E+01	1.3379E+01	5.3218E+01	5.1373E-05	1.1397E-01
13	1.0318E+01	1.3592E+01	5.3269E+01	5.2955E-05	1.0293E-01
14	1.0186E+01	1.3768E+01	5.3338E+01	5.4325E-05	9.5268E-02
15	1.0073E+01	1.3914E+01	5.3421E+01	5.5519E-05	9.0283E-02

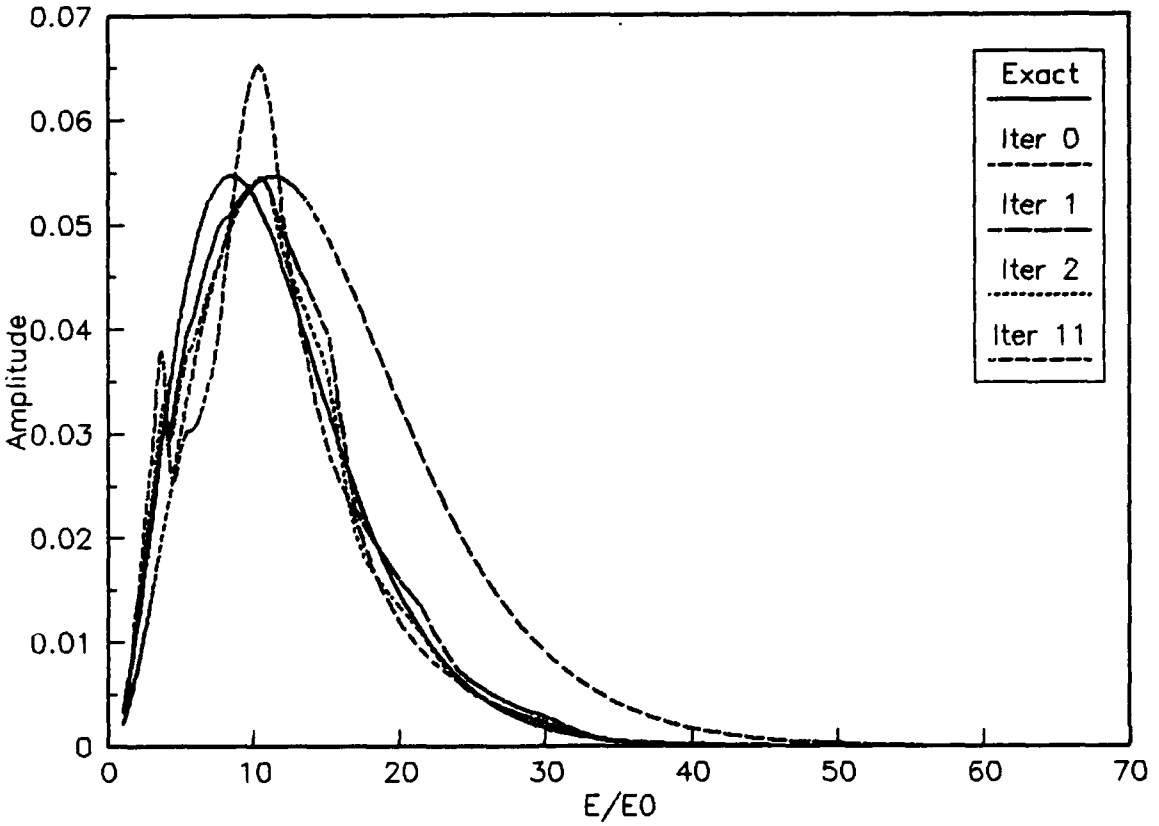


Figure 12: Case TC1.Ø (smoother)

It is important to note the L_r does not exhibit the local minimum characteristic of L_a. This results from large relative errors at the tail of the unfolded spectrum and is an artifact of the scaling process. The error is seen most clearly in Figure 13.

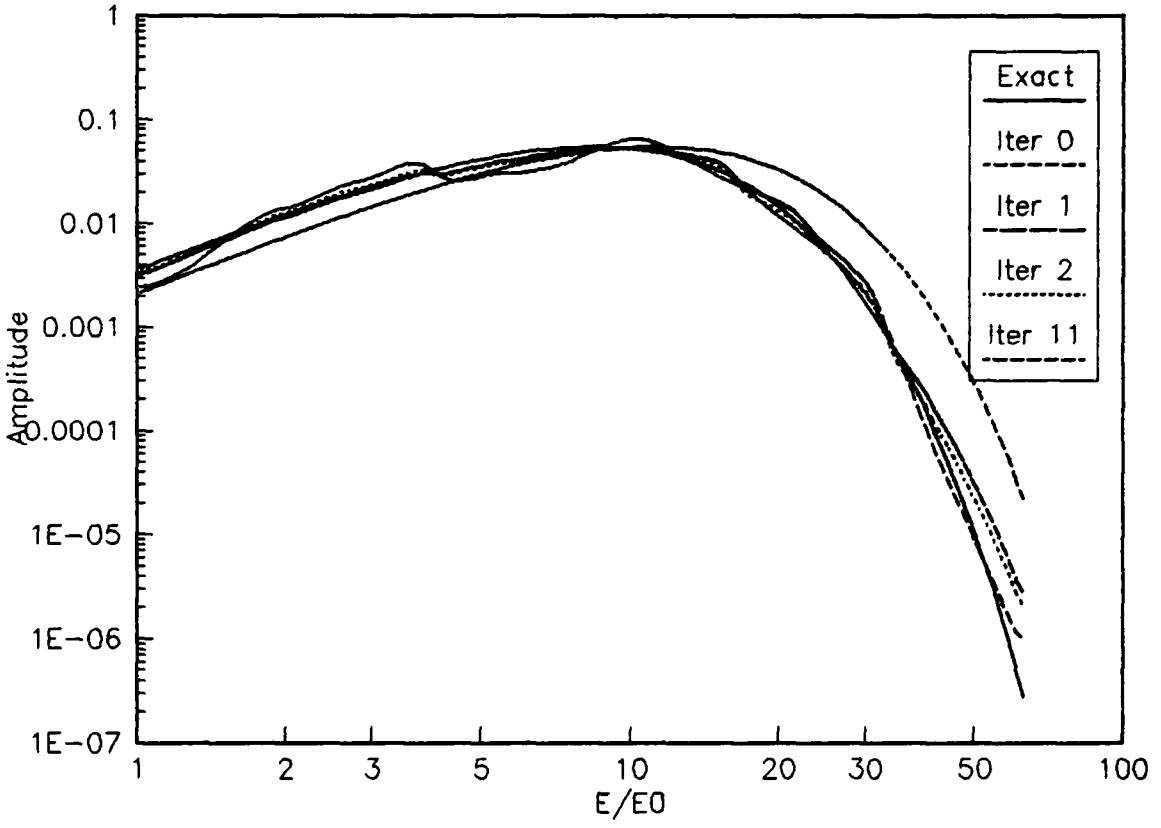


Figure 13: Case TC1.0 (smoother on)

In case TC1.1s, the exact and guessed spectra are identical to case TC1.1. In this case the smoother was included. The results are contained in Table 8. Table 8 shows that all the test statistics, except χ^2, exhibit minimum at zero iterations. Figure 14 shows the unfolded spectra at various iterations.

Table 8: Test Statistics for Case TC1.1 (smoother on)

Iter	χ^2	χ^2_I	Q_l	L_a	L_r
0	3.0559E+01	0.0000E+00	3.0559E+01	0.0000E+00	0.0000E+00
1	1.9571E+01	2.8438E+00	4.3084E+01	5.4574E-06	4.2492E-03
2	1.6210E+01	5.1173E+00	4.5264E+01	1.2776E-05	1.0494E-02
3	1.4293E+01	7.0836E+00	4.5550E+01	2.0097E-05	1.7562E-02
4	1.3114E+01	8.6578E+00	4.5842E+01	2.6567E-05	2.4673E-02
5	1.2337E+01	9.8828E+00	4.6206E+01	3.2045E-05	3.1529E-02
6	1.1796E+01	1.0831E+01	4.6657E+01	3.6619E-05	3.8032E-02
7	1.1400E+01	1.1567E+01	4.7148E+01	4.0431E-05	4.4164E-02
8	1.1099E+01	1.2142E+01	4.7654E+01	4.3619E-05	4.9944E-02
9	1.0861E+01	1.2596E+01	4.8156E+01	4.6299E-05	5.5403E-02
10	1.0668E+01	1.2957E+01	4.8643E+01	4.8566E-05	6.0573E-02

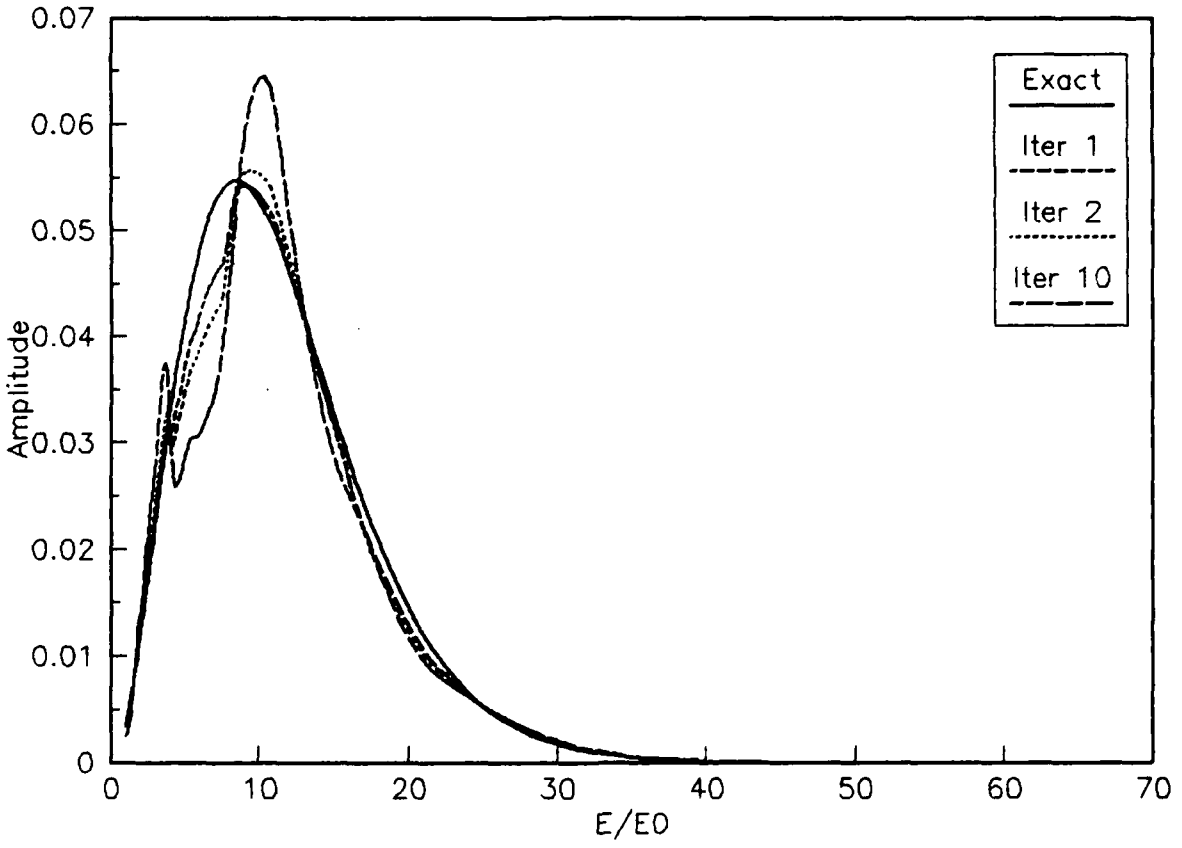


Figure 14: Case TC1.1 (smoother on)

The entire analysis process for case TC1 was repeated for an additional simulated noise set. The new noisy set of signals were generated by reseeding the random number generator with a different seed. The results of the effort are tabulated in Appendix B.

Results of Test Case 2 (TC2). In the series of test cases prefixed TC2, the exact spectrum was changed to a two temperature Planckian of temperatures 1.2 E_0^1 and 6 E_0^1.

For case TC2.0, a one temperature Planckian of temperature of 3 E_0^1 and scaling of 2.5 was used as the first guess. Additionally, the smooth routine was deactivated, leaving only the fit process in the iterative unfolder. Table 9 shows the results for this case. Note that while χ^2 is decreasing monotonically; χ_f^2, Q_l, L_a, and L_r all exhibit minimums. Figure 15 shows the unfolded spectra at the iteration numbers corresponding to these minimums.

Table 9: Test Statistics for Case TC2.0 (smoother off)

Iter	χ^2	χ_f^2	Q_l	L_a	L_r
0	3.2298E+02	3.3139E+02	3.2298E+02	4.7672E-03	6.1341E-01
1	1.2831E+02	7.2915E+01	2.0727E+02	1.1621E-03	2.3632E-01
2	1.6717E+01	7.8585E+00	5.2088E+01	2.7262E-04	1.2876E-01
3	7.2972E+00	1.1325E+01	3.1950E+01	4.2788E-04	1.3393E-01
4	5.4568E+00	1.4874E+01	2.6839E+01	5.8085E-04	1.4216E-01
5	4.8403E+00	1.6956E+01	2.5199E+01	6.8736E-04	1.4897E-01
6	4.5272E+00	1.8178E+01	2.4646E+01	7.6405E-04	1.5466E-01
7	4.3175E+00	1.8935E+01	2.4490E+01	8.2356E-04	1.5962E-01
8	4.1517E+00	1.9434E+01	2.4490E+01	8.7315E-04	1.6412E-01
9	4.0082E+00	1.9784E+01	2.4552E+01	9.1702E-04	1.6831E-01
10	3.8786E+00	2.0046E+01	2.4639E+01	9.5765E-04	1.7232E-01

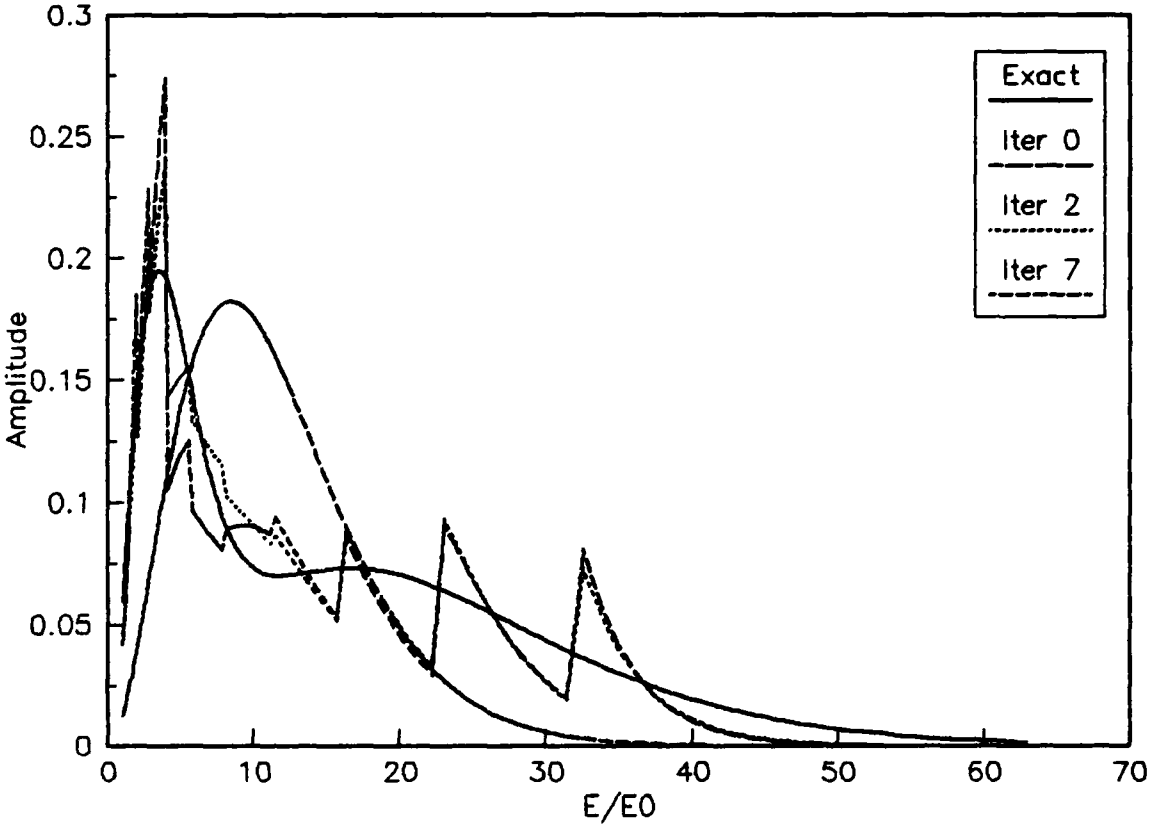


Figure 15: Case TC2.0 (smoother off)

In case TC2.1, the guess of TC2.0 was revised. In this attempt, a two temperature guess was used instead of the one temperature guess. The revised guess was a Planckian of temperatures 1.8 E_i^0 and 5.4 E_i^0 with scaling of 1.25 and 2.8, respectively. Note that in Table 10 χ^2 still decreases monotonically, giving no information, but all of the other statistics show minimums. Figure 16 shows the corresponding unfolded spectra at the iteration numbers at which these minimums occurred.

Table 10: Test Statistics for Case TC2.1 (smoother off)

Iter	χ^2	χ^2_f	Q_t	L_a	L_r
0	2.8875E+02	2.0703E+02	2.8875E+02	2.6977E-03	3.3271E-01
1	3.8268E+01	9.2843E+00	7.2204E+01	1.9585E-04	3.3001E-02
2	1.3795E+01	5.9600E+00	4.1710E+01	1.9680E-04	2.4710E-02
3	9.8565E+00	1.0243E+01	3.6765E+01	3.6150E-04	3.4008E-02
4	8.5773E+00	1.3033E+01	3.5618E+01	4.8439E-04	4.3276E-02
5	7.8890E+00	1.4693E+01	3.5403E+01	5.7113E-04	5.1445E-02
6	7.4044E+00	1.5736E+01	3.5500E+01	6.3628E-04	5.8767E-02
7	7.0137E+00	1.6439E+01	3.5701E+01	6.8887E-04	6.5509E-02
8	6.6760E+00	1.6946E+01	3.5928E+01	7.3413E-04	7.1862E-02
9	6.3740E+00	1.7334E+01	3.6151E+01	7.7517E-04	7.7946E-02
10	6.0988E+00	1.7648E+01	3.6362E+01	8.1384E-04	8.3841E-02

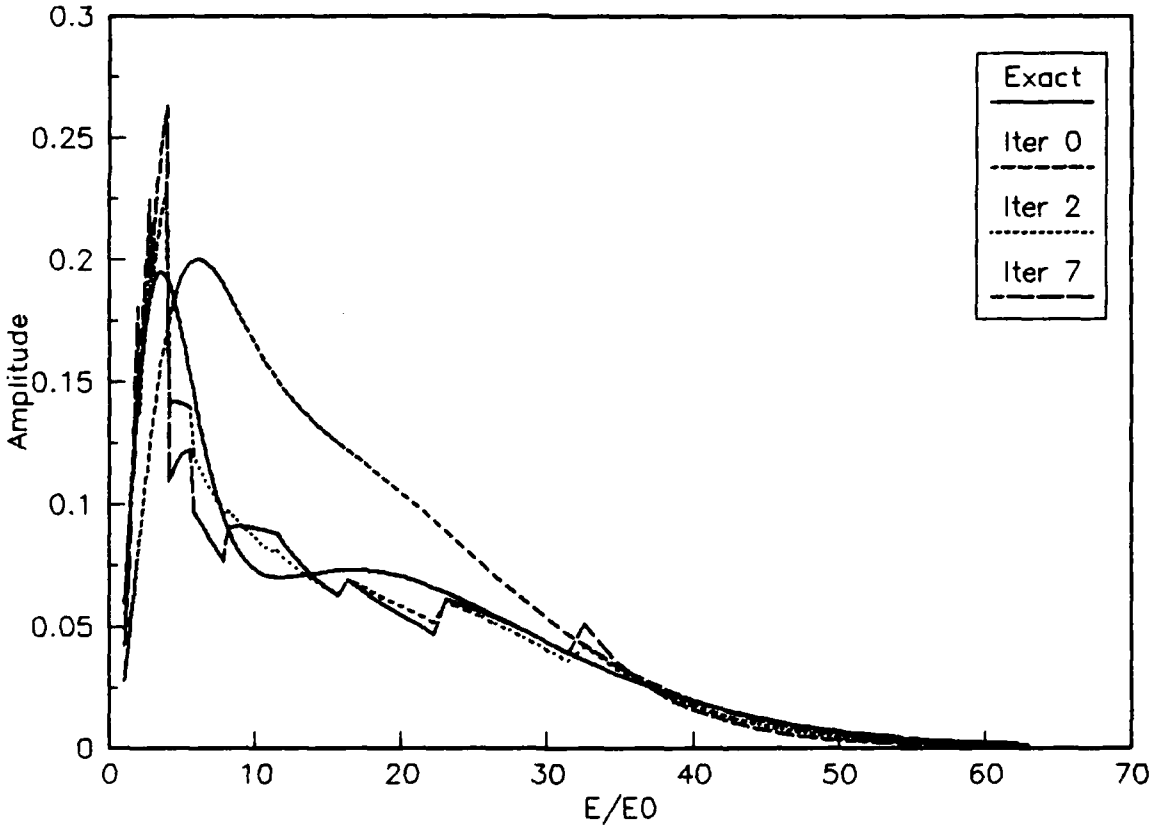


Figure 16: Case TC2.1 (smoother off)

Case TC2.2, uses the exact spectrum as the guess. Additionally, the iterative unfolder uses only the fit routine. Table 11 shows the results for this case. Note that all the test statistics exhibit minimums at zero iterations.

Table 11: Test Statistics for Case TC2.2 (smoother off)

Iter	χ^2	χ^2_s	Q_t	L_a	L_r
0	3.0559E+01	0.0000E+00	3.0559E+01	0.0000E+00	0.0000E+00
1	1.5790E+01	4.8349E+00	3.5080E+01	1.0569E-04	6.7420E-03
2	1.2261E+01	8.7125E+00	3.6997E+01	2.3746E-04	1.5049E-02
3	1.0614E+01	1.1399E+01	3.8063E+01	3.4523E-04	2.3100E-02
4	9.6325E+00	1.3187E+01	3.8803E+01	4.2777E-04	3.0535E-02
5	8.9374E+00	1.4401E+01	3.9352E+01	4.9193E-04	3.7435E-02
6	8.3877E+00	1.5258E+01	3.9772E+01	5.4389E-04	4.3940E-02
7	7.9238E+00	1.5888E+01	4.0102E+01	5.8809E-04	5.0174E-02
8	7.5174E+00	1.6373E+01	4.0369E+01	6.2753E-04	5.6226E-02
9	7.1534E+00	1.6763E+01	4.0593E+01	6.6422E-04	6.2153E-02
10	6.8228E+00	1.7089E+01	4.0787E+01	6.9944E-04	6.7991E-02

Cases TC2.0s-TC2.2s are a repeat of Cases TC2.0-TC2.2, except the smooth routine was added to the iterative unfold process.

In case TC2.0s, the guessed spectrum is the same one temperature Planckian used in case TC2.0. Table 12 shows that all the test statistics exhibit minimums. Figure 17 shows the unfolded spectra corresponding to the iteration number at the minimums. Note that comparisons of Tables 9 and 12 show that the smoothing additions enhance the speed of fit (χ^2) and deepens the minimum of χ^2_s.

Table 12: Test Statistics for Case TC2.0 (smoother on)

Iter	χ^2	χ^2_s	Q_t	L_a	L_r
0	3.2298E+02	3.3139E+02	3.2298E+02	4.7672E-03	6.1341E-01
1	1.3047E+02	7.5116E+01	2.0841E+02	1.1032E-03	2.1902E-01
2	1.8355E+01	7.4978E+00	5.2745E+01	1.8141E-04	1.0569E-01
3	8.8323E+00	9.9861E+00	3.3005E+01	2.9344E-04	1.0419E-01
4	7.0802E+00	1.2911E+01	2.8446E+01	4.0967E-04	1.0693E-01
5	6.5962E+00	1.4621E+01	2.7354E+01	4.8489E-04	1.0904E-01
6	6.4284E+00	1.5573E+01	2.7342E+01	5.3344E-04	1.1049E-01
7	6.3644E+00	1.6125E+01	2.7723E+01	5.6641E-04	1.1151E-01
8	6.3398E+00	1.6454E+01	2.8255E+01	5.9012E-04	1.1227E-01
9	6.3305E+00	1.6656E+01	2.8839E+01	6.0819E-04	1.1287E-01
10	6.3267E+00	1.6783E+01	2.9434E+01	6.2275E-04	1.1338E-01

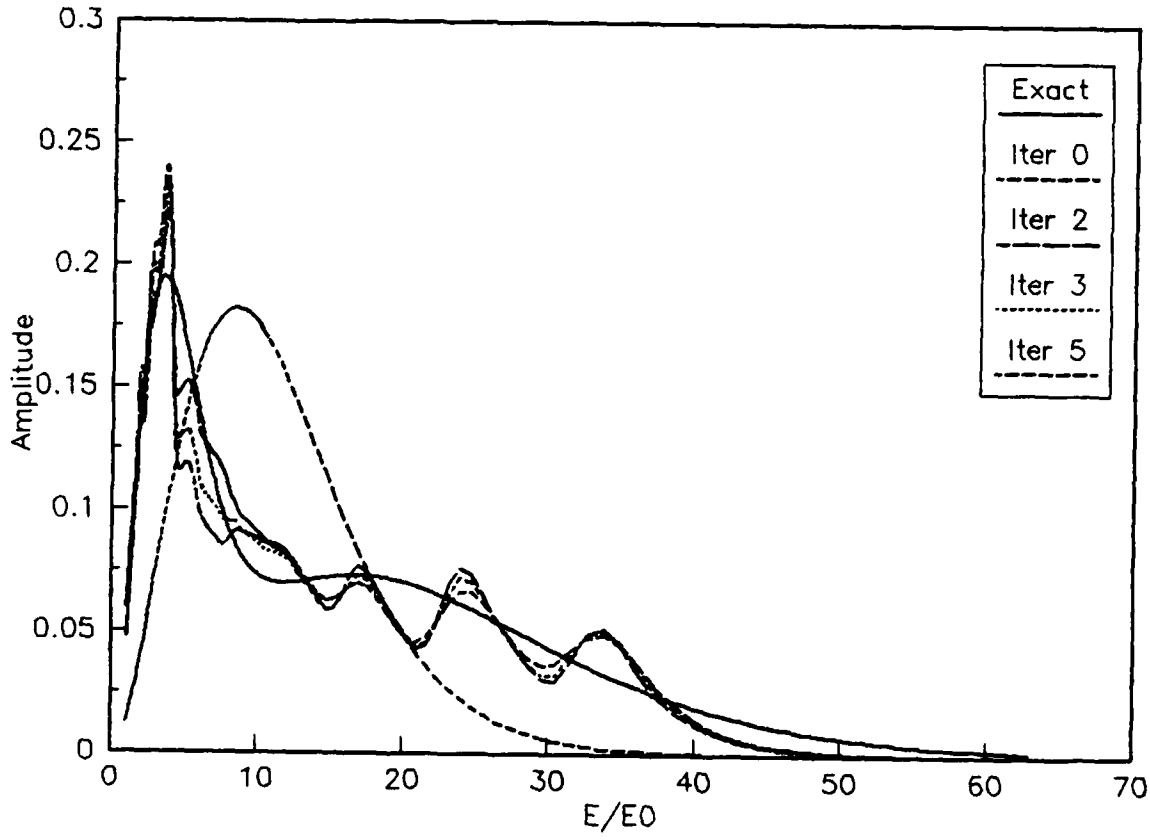


Figure 17: Case TC2.0 (smoother on)

In case TC2.1s, the guessed spectrum is changed to the two temperature Planckian used in case TC2.1. Table 13 shows the results for this case. In Table 13, it is seen that all the test statistics exhibit minimums except χ^2. Figure 18 depicts the unfolded spectra at these minimums.

Table 13: Test Statistics for Case TC2.1 (smoother on)

Iter	χ^2	χ^2_I	Q_I	L_a	L_r
0	2.8875E+02	2.0703E+02	2.8875E+02	2.6977E-03	3.3271E-01
1	3.9158E+01	9.5276E+00	7.2261E+01	1.8386E-04	3.1404E-02
2	1.4747E+01	5.5040E+00	4.2209E+01	1.5847E-04	2.0852E-02
3	1.0862E+01	9.3273E+00	3.7699E+01	2.9709E-04	2.7977E-02
4	9.6556E+00	1.1831E+01	3.6939E+01	3.9703E-04	3.5130E-02
5	9.0584E+00	1.3302E+01	3.7061E+01	4.6284E-04	4.1224E-02
6	8.6762E+00	1.4208E+01	3.7471E+01	5.0789E-04	4.6462E-02
7	8.3933E+00	1.4800E+01	3.7967E+01	5.4043E-04	5.1085E-02
8	8.1652E+00	1.5209E+01	3.8474E+01	5.6518E-04	5.5261E-02
9	7.9720E+00	1.5505E+01	3.8962E+01	5.8495E-04	5.9102E-02
10	7.8036E+00	1.5728E+01	3.9422E+01	6.0144E-04	6.2682E-02

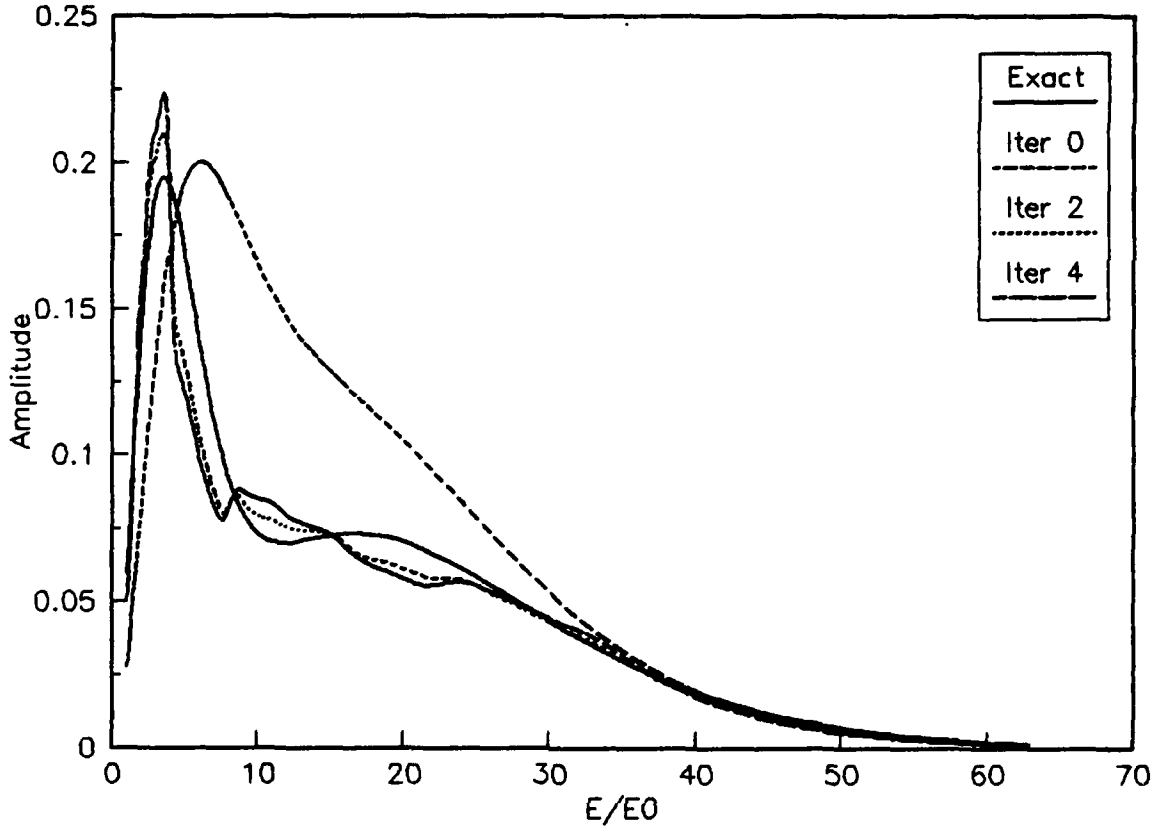


Figure 18: Case TC2.1 (smoother on)

In case TC2.2s, the guessed spectra was changed to the exact spectrum as in case TC2.2. This time the smoother was implemented. Again, all statistics exhibit minimums at zero iterations.

Table 14: Test Statistics for Case TC2.2 (smoother on)

Iter	χ^2	χ^2_i	Q_i	L_a	L_r
0	3.0559E+01	0.0000E+00	3.0559E+01	0.0000E+00	0.0000E+00
1	1.6193E+01	4.6110E+00	3.5050E+01	1.0049E-04	6.2122E-03
2	1.2811E+01	8.1976E+00	3.7187E+01	2.2182E-04	1.3542E-02
3	1.1265E+01	1.0634E+01	3.8460E+01	3.1820E-04	2.0401E-02
4	1.0375E+01	1.2234E+01	3.9423E+01	3.8938E-04	2.6511E-02
5	9.7764E+00	1.3303E+01	4.0190E+01	4.4197E-04	3.1972E-02
6	9.3284E+00	1.4044E+01	4.0819E+01	4.8174E-04	3.6929E-02
7	8.9703E+00	1.4574E+01	4.1347E+01	5.1276E-04	4.1501E-02
8	8.6720E+00	1.4967E+01	4.1800E+01	5.3780E-04	4.5774E-02
9	8.4169E+00	1.5267E+01	4.2199E+01	5.5872E-04	4.9808E-02
10	8.1951E+00	1.5505E+01	4.2556E+01	5.7675E-04	5.3643E-02

The series of cases prefixed TC2 were repeated using a new set of noisy signals. The results are tabulated in Appendix C.

Summary of Test Results. The following table is a collection of the results from cases TC1 and TC2, discussed above, and the cases tabulated in Appendices B and C. In Table 15, the absolute error in the unfolded spectral approximation is tabulated for the various cases and stoppage criteria.

Table 15: Absolute Error for Various Stoppage Criteria

Case	$\chi^2 - n_t$	$\chi^2 - n_t/2$	Q_t
TC1.0	1.95E-05	6.66E-05	3.37E-05
TC1.0^s	1.80E-05	5.96E-05	4.95E-05
TC1.0*	1.02E-05	4.30E-05	8.82E-06
TC1.0^s*	9.27E-06	3.14E-05	8.04E-06
TC2.0	2.73E-04	4.27E-04	8.24E-04
TC2.0^s	2.93E-04	4.10E-04	5.33E-04
TC2.0**	4.79E-04	6.79E-04	3.66E-04
TC2.0^s**	2.76E-04	6.58E-04	2.76E-04
TC2.1	1.97E-04	5.71E-04	5.71E-04
TC2.1^s	1.58E-04	5.41E-04	3.97E-04
TC2.1**	2.23E-04	7.92E-04	2.23E-04
TC2.1^s**	2.07E-04	8.58E-04	2.07E-04

^{s} denotes smoother on
* results contained in Appendix B
** results contained in Appendix C

Case TC2.0 (smoother on) was repeated several times while randomly varying the amounts of noise added to the ideal signals. The absolute error of the unfolded approximation to its exact counterpart was tabulated. Table 16 shows the results from ten different sets of noisy signals.

Table 16: Absolute Error for Different Noise Sets on Case TC2.0 (smoother on)

Attempt	$\chi^2 - n_t$	$\chi^2 - n_t/2$	Q_t
1	1.10E-03	4.10E-04	5.33E-04
2	2.76E-04	6.58E-04	2.76E-04
3	1.97E-04	4.77E-04	2.80E-04
4	2.30E-04	1.67E-04	1.61E-04
5	1.44E-04	1.70E-04	1.42E-04
6	4.74E-04	2.79E-04	2.79E-04
7	1.41E-04	3.57E-04	1.49E-04
8	2.91E-04	1.78E-04	1.59E-04
9	3.49E-04	3.37E-04	3.76E-04
10	2.55E-04	2.31E-04	2.31E-04

V. Summary and Conclusions

Summary

Validation. The validation consisted of a two step process. First, the iterative unfold routine was validated. Validation cases one through three showed that the iterative unfold routine operated as expected. Both the fit routine alone and fit/smooth routines in tandem produced spectra that converge toward a spectrum that produces measured signals as iteration increase.

Second, the cross-validation phase was validated. Two ideal cases (no measurement error) were constructed to test the proper operation of cross-validation. The first case showed that cross-validation predicted no modification to the trial spectrum when it corresponds to the exact spectrum. The second case showed that cross-validation produced a monotonically decreasing Q_i function when the guessed spectrum is still becoming a better approximation to the exact spectrum.

Test Cases. Two basic test cases were constructed. Test case 1 used an exact spectrum equal to a one temperature Planckian. Each variation of test case 1 was prefixed with TC1. Test case 2 used a two temperature Planckian as the exact spectrum. Similarly, each variation of test case 2 was prefixed with TC2. In each case, the guessed spectrum, the simulated noise set, and the use of the iterative unfolder (fit or fit/smooth) were varied.

Conclusions

In each test case attempted, cross-validation predicted an optimum iteration number. Although the smooth routine competes with the fit routine, each test showed a deeper minimum for χ^2_f when the smooth routine was used. As a result, smoothing out k-edge discontinuities resulted in an overall better fit to the exact spectrum.

Four cases used the exact spectrum as the guessed spectrum (i.e. guessed the right answer). In each of these cases, cross-validation predicted no modification to the guessed spectrum. In contrast, a prediction based on any reasonable value of χ^2 lead to overfitting of the data. Typically, the stopping values used for χ^2 (number of degrees of freedom) are less than the number of instruments used and may be allowed to fall to quite small values.

In the twelve test cases depicted in Table 15, the smoother implementation, noisy measured signal set, and predicted spectra were varied. For those cases the use of any value of χ^2 less than seventeen (the number of instruments) would lead to data overfit. In fact, a choice of χ^2 equal to seventeen would be the best fit in eight of twelve cases. However, the stopping point predicted by minimization Q_i lead to the best fit in six of twelve cases.

The multiple Planckian case (TC2.0) was used to study the effects of noise on cross-validations ability to stop the iterative process. Noise was randomly generated and added to the ideal signals. Again, the absolute error was compared

for the various stopping criteria. Table 16 showed that using χ^2 equal to 17.0 provided the best predictor in only three out of ten cases, while a value 8.5 provided the best estimator in four of ten cases. This result is quite different from that of Table 15, where a χ^2 value of 17.0 was clearly the best choice. Cross-validation provided the best estimator in five of ten cases. In the ten cases attempted cross-validation overfit the data five times, predicted the best fit three times, and under fit the data twice. On average, cross-validation tended to overfit the data by about one iteration. In contrast, using χ^2 equal to the number of instruments under fit the data in seven out of ten cases, and averaged slightly more than one iteration underfit. Using χ^2 equal to half the instruments lead to overfit of the data in eight of ten cases, and averaged about four iterations overfit.

In conclusion, the absolute error associated with choosing any value of χ^2 as the stopping point appears to be problem dependent. Consequently, there is some question as what value of χ^2 to use. Clearly, very small values lead to data overfit. Over the range of cases attempted, cross-validation and the minimization of Q_i lead to a slight overfit of data (about one iteration on average) but provided a less judgement and problem dependent approach to stopping the iterative unfold method.

VI. Recommendations

The following are recommendations for study in this area. First, the use of cross-validation should be explored to predict other parameters for the iterative unfold technique for deconvolution. Some possible parameters are types of smoothing kernels, types of instrument weighting functions (fit algorithm), and energy-bin spacing. Second, this study should be continued to explore the use of cross-validation as a parameter predictor for the basis function method. Finally, a third area of study should consist of exploration of the use of generalized cross-validation as a "stand-alone" deconvolution process.

Appendix A: Source Code for CvIter.bas

The following is the source code from CVIter.bas which is a Microsoft QuickBASIC V4.5 computer code used generate the data used is this study.

```
'*******************************************************************************
' Program to Unfold by Iteration Method Using Cross-validation
'
'     Program Name: CVIter.bas
'     Version:  1.0
'     Author:  Dennis J. Miller
'
'
'     Comments:  Partial Iteration Steps (FIT), 1/Yu weighting addition (FIT),
'                vote
'*******************************************************************************
'
'  Variable Name        Description
'  -------------        -----------
'  datestring$          date of program execution
'  fexact$              file name containing exact spectrum
'  filename$            file(s) name containing exact spectra
'  fpred$               file name containing predicted spectrum
'  funfold$             file name containing unfolded spectrum
'  SetHardCopy$         flag for hard copy output (Y or N)
'  SetNoise$            flag for simulated noise (Y or N)
'  SetSmoother$         flag to turn on smoother (Y or N)
'  SetValidation$       flag to trun on cross-validation (Y or N)
'  timestring$          time of program execution
'  version$             program version number
'
'  FinishTime           finish time of each iteration
'  Chi                  Chi^2 (measured signals)
'  ChiExact             Chi^2 (exact signals)
'  CumTime              cummunlative program execution time
'  sig                  fractional instrument uncertainty
'  StartTime            start time of each iteration

'  AminQ                place holder
'  iteropt              optimum iteration number as chosen by cross-validation
'  nbins                number of energy bins
'  ninst                number of instruments
'  niter                max number of iterations
'*******************************************************************************

DEFINT I-N
DEFSNG A-H, O-Z
```

```
DECLARE FUNCTION ChiSqd (InstOmit, su(), b())
DECLARE FUNCTION Qterm (Iomit, b(), su())
DECLARE FUNCTION SpectralNorm (su(), ntype)
DECLARE FUNCTION Pd (x, ave, sig)

DECLARE SUB Fit (n1, n2, su())
DECLARE SUB FileOutput (index)
DECLARE SUB GenNoise (epsilon(), bexact(), b(), sig)
DECLARE SUB GenRFs (sp(), e0(), e1(), r(), dE(), Yp())
DECLARE SUB HardCopy (index)
DECLARE SUB header ()
DECLARE SUB MeasToPred (b())
DECLARE SUB Sinput ()
DECLARE SUB Spectrum (filename$, s())
DECLARE SUB Setweights (SpaceType$, weight())
DECLARE SUB Smooth (su(), weight())
DECLARE SUB vote (Q1(), iteropt1, icounter)

version$ = "1.0"
datestring$ = DATE$
timestring$ = TIME$

CumTime = 0

'Set parameters for unfold

CALL Sinput

' These are global variables; the subroutine that first uses them is listed

REDIM a(1 TO ninst)                    'dummy matrix to calculate H
REDIM b(1 TO ninst)                    'measured-to-predicted ratio w/noise
REDIM bexact(1 TO ninst)               'measured-to-predicted ratio (exact)
REDIM c(1 TO ninst)                    'unfolded-to-predicted ratio
REDIM d(1 TO ninst)                    'integral of R*Sp*dE
REDIM e1(1 TO ninst)                   'k-edge of filters
REDIM e0(1 TO ninst)                   'k-edge of fluoresors
REDIM Qloss(0 TO niter)                'Loss function Q vector
REDIM Q1(1 TO ninst, 0 TO niter)       'Loss Function 1 instrument omitted
REDIM r(1 TO ninst, 1 TO nbins)        'response function of a detector for an
REDIM sa(1 TO nbins)                   'energies of the actual spectrum
REDIM sigma(1 TO ninst)                'variation of instrument measurement
REDIM sp(1 TO nbins)                   'energies of the predicted spectrum
REDIM su(1 TO nbins)                   'energies of the unfolded spectrum
REDIM suguess(1 TO nbins)              'first guess at the unfolded spectrum
REDIM weight(1 TO nbins, 1 TO nbins)   'relative weights of surrounding bins
REDIM dE(1 TO nbins)                   'energy bin width
REDIM epsilon(1 TO ninst)              'random noise added to signal
REDIM Yp(1 TO ninst)                   'predicted signals

'  Retrieve actual, predicted, and initial guess at unfolded spectrum

   CALL Spectrum(fpred$, sp())
   CALL Spectrum(fexact$, sa())
   CALL Spectrum(funfold$, suguess())
```

```
' Generate Response Data

   CALL GenRFs(sp(), e0(), e1(), r(), dE(), Yp())

' Set Weights for Smoother

   CALL Setweights(SpaceType$, weight())

' Calculate measured to predicted ratio

   CALL MeasToPred(bexact())

' Generate Random Noise

   IF UCASE$(LEFT$(SetNoise$, 1)) = "Y" THEN
      CALL GenNoise(epsilon(), bexact(), b(), sig)
   ELSE
      FOR inst = 1 TO ninst
         b(inst) = bexact(inst)
      NEXT inst
   END IF

   FOR inst = 1 TO ninst
      sigma(inst) = sig * b(inst)
   NEXT inst

' Generate Hard Copy Output

   FileOutput (1)
   IF SetHardCopy$ = "Y" THEN CALL HardCopy(1)

' Start Cross Validation (omit one instrument)

IF SetValidation$ = "Y" THEN

   FOR inst = 1 TO ninst

      FOR ibin = 1 TO nbins
         su(ibin) = suguess(ibin)
      NEXT ibin

'     Calculate Qterm for 0th iteration (for guess)

      FileOutput (7)
      IF SetHardCopy$ = "Y" THEN CALL HardCopy(7)

      Q1(inst, 0) = Qterm(inst, b(), su())
      Qloss(0) = Qloss(0) + Q1(inst, 0)
      Chi = ChiSqd(inst, su(), b())
      ChiExact = ChiSqd(inst, su(), bexact())

      iter = 0
      FileOutput (8)
      IF SetHardCopy$ = "Y" THEN CALL HardCopy(8)
```

```
      FOR iter = 1 TO niter

         StartTime = TIMER

         CALL Fit(inst, 0, su())
         IF SetSmoother$ = "Y" THEN CALL Smooth(su(), weight())

         Q1(inst, iter) = Qterm(inst, b(), su())
         Chi = ChiSqd(inst, su(), b())
         ChiExact = ChiSqd(inst, su(), bexact())

         FileOutput (4)
         IF SetHardCopy$ = "Y" THEN CALL HardCopy(4)

         FinishTime = TIMER
         CumTime = CumTime + (FinishTime - StartTime) / 60

         LOCATE 1, 1
            PRINT "Cross-Validation Currently Underway..."
         LOCATE 15, 23
            PRINT USING "Instrument currently omitted: ## "; inst
         LOCATE 16, 26
            PRINT USING "Iteration Number: ## of ##"; iter; niter
         LOCATE 17, 22
            PRINT USING "Qterm this iteration:  ##.####^^^^"; Q1(inst, iter)
         LOCATE 18, 22
            PRINT USING "Chi-Squared (Exact):   ##.####^^^^"; ChiExact
         LOCATE 19, 22
            PRINT USING "     Chi-Squared:  ##.####^^^^    "; Chi
         LOCATE 20, 28
            PRINT USING "Run time = ###.### min"; CumTime

      NEXT iter
      IF SetHardCopy$ = "Y" THEN LPRINT

   NEXT inst
   IF SetHardCopy$ = "Y" THEN LPRINT

   FOR inst = 1 TO ninst
      FOR iter = 1 TO niter
         Qloss(iter) = Qloss(iter) + Q1(inst, iter)
      NEXT iter
   NEXT inst

   CLS

   AminQ = 1E+09
   FOR iter = 0 TO niter
      IF Qloss(iter) < AminQ THEN
         AminQ = Qloss(iter)
         iteropt = iter
      END IF
   NEXT iter

END IF
```

```
' Unfold Spectrum for optimum value

   FOR ibin = 1 TO nbins
      su(ibin) = suguess(ibin)
   NEXT ibin
   Chi = ChiSqd(0, su(), b())
   ChiExact = ChiSqd(0, su(), bexact())

   AbsNorm = SpectralNorm(su(), 0)
   RelNorm = SpectralNorm(su(), 1)

   FileOutput (9)
   FileOutput (5)
   IF SetHardCopy$ = "Y" THEN CALL HardCopy(9)
   IF SetHardCopy$ = "Y" THEN CALL HardCopy(5)

   IF SetValidation$ = "N" THEN iteropt = niter
   IF iteropt >= 0 THEN

      FOR iter = 1 TO niter

         filename$ = fout$ + LTRIM$(STR$(iter)) + ".asc"
         OPEN filename$ FOR OUTPUT AS #1

         CALL Fit(0, 0, su())
         IF SetSmoother$ = "Y" THEN CALL Smooth(su(), weight())

         Chi = ChiSqd(0, su(), b())
         ChiExact = ChiSqd(0, su(), bexact())

         AbsNorm = SpectralNorm(su(), 0)
         RelNorm = SpectralNorm(su(), 1)

         LOCATE 1, 1
            PRINT "Unfolding Spectrum..."
         LOCATE 15, 30
            PRINT USING "Iteration Number ## of ##:"; iter; niter
         LOCATE 16, 30
            PRINT USING "   Chi^2 = ##.####^^^^    "; Chi

         FileOutput (6)
         IF SetHardCopy$ = "Y" THEN CALL HardCopy(6)

         FOR ibin = 1 TO nbins
            PRINT #1, su(ibin)
         NEXT ibin
         CLOSE #1

         LOCATE 17, 30
            PRINT "Saving Output to "; filename$

      NEXT iter
   END IF
```

```
      FileOutput (3)
      IF SetHardCopy$ = "Y" THEN CALL HardCopy(3)

END

FUNCTION ChiSqd (InstOmit, su(), b()) STATIC

' ============================================================
' ChiSqd calculates the conventional Chi^2 (exact or measured) based on the
' unfolded spectra
' ============================================================

   SHARED sigma(), r(), dE()

   ninst = UBOUND(r, 1)
   nbins = UBOUND(r, 2)

   REDIM c(1 TO ninst)

   FOR inst = 1 TO ninst

      c(inst) = 0
      FOR ibin = 1 TO nbins
         c(inst) = c(inst) + r(inst, ibin) * su(ibin) * dE(ibin)
      NEXT ibin

   NEXT inst

   TChiSqd = 0
   FOR inst = 1 TO ninst
      IF InstOmit <> inst THEN
         TChiSqd = TChiSqd + ((b(inst) - c(inst)) / sigma(inst)) ^ 2
      END IF
   NEXT inst
   ChiSqd = TChiSqd

END FUNCTION

SUB FileOutput (index)

' ============================================================
' FileOutput copies output to the specified output file
' ============================================================

   SHARED e0(), e1(), nopen, nclosed, bexact(), numinst
   SHARED b(), epsilon(), Q1(), Qloss(), iteropt, iteropt1
   SHARED inst, Chi, ChiExact, iter, fpred$, fexact$
   SHARED SetSmoother$, version$, datestring$, timestring$
   SHARED AbsNorm, RelNorm, fout1$

   ninst = UBOUND(b, 1)
   niter = UBOUND(Q1, 2)

   IF index = 1 THEN
```

```
      OPEN fout1$ FOR OUTPUT AS #2

      PRINT #2, "Version;  "; version$
      PRINT #2, "Date:  "; datestring$
      PRINT #2, "Time:  "; timestring$
      PRINT #2, "Author:  Dennis Miller"
      PRINT #2,
      PRINT #2, "File for Exact Spectrum:  "; fexact$
      PRINT #2, "File for Predicted Spectrum:  "; fpred$
      PRINT #2,
      PRINT #2, "Smoother ON:  "; SetSmoother$
      PRINT #2,
      PRINT #2,

      PRINT #2, "inst    edge 1       edge 2    "
      FOR inst = 1 TO ninst
         IF inst <= nopen THEN
            PRINT #2, USING "##     ##.#####     ##.#####"; (inst); e0(inst); e1(inst)
         ELSE
            PRINT #2, USING "##     ##.#####     "; (inst); e0(inst)
         END IF
      NEXT inst
      PRINT #2,

      PRINT #2, " inst      bexact()       b()           epsilon()"
      FOR inst = 1 TO ninst
         PRINT #2, USING " ##       ##.###^^^^     ##.###^^^^     ##.###^^^^"; inst;
bexact(inst); b(inst); epsilon(inst)
      NEXT inst
      PRINT #2,

      CLOSE #2

   ELSEIF index = 2 THEN

   ELSEIF index = 3 THEN

      OPEN fout1$ FOR APPEND AS #2

      PRINT #2,
      PRINT #2, "The optimum iteration number (minimization of Qloss): "; iteropt
      PRINT #2,

      CLOSE #2

   ELSEIF index = 4 THEN
```

```
      OPEN fout1$ FOR APPEND AS #2

      PRINT #2, USING "##     ##    ##.####^^^^    ##.####^^^^     ##.####^^^^"; iter;
inst; Q1(inst, iter); Chi; ChiExact

      CLOSE #2

   ELSEIF index = 5 THEN

      OPEN fout1$ FOR APPEND AS #2

      PRINT #2, USING "  0   ##.####^^^^   ##.####^^^^  ##.####^^^^   ##.####^^^^
##.####^^^^"; Chi; ChiExact; Qloss(0); AbsNorm; RelNorm

      CLOSE #2

   ELSEIF index = 6 THEN

      OPEN fout1$ FOR APPEND AS #2

      PRINT #2, USING " ##   ##.####^^^^   ##.####^^^^  ##.####^^^^   ##.####^^^^
##.####^^^^"; iter; Chi; ChiExact; Qloss(iter); AbsNorm; RelNorm

      CLOSE #2

   ELSEIF index = 7 THEN

      OPEN fout1$ FOR APPEND AS #2

      PRINT #2,
      PRINT #2, "Iter   IO     Qterm          Chi^2          Chi^2 Exact"

      CLOSE #2

   ELSEIF index = 8 THEN

      OPEN fout1$ FOR APPEND AS #2

      PRINT #2, USING "##     ##    ##.####^^^^     ##.####^^^^     ##.####^^^^"; iter;
inst; Q1(inst, iter); Chi; ChiExact

      CLOSE #2

   ELSEIF index = 9 THEN

      OPEN fout1$ FOR APPEND AS #2

      PRINT #2,
      PRINT #2, "Iter   Chi^2        Chi^2 Exact  Qloss        AbsNorm       RelNorm"

      CLOSE #2

   END IF

END SUB
```

```
SUB Fit (n1, n2, su()) STATIC

' ============================================================
' Fit is the fitting routine used in the iterative unfold method
' ============================================================

SHARED b(), sp(), sa(), r(), dE(), Yp(), e0(), e1()

v = 1          'used for partial iterative steps

MUM = 1000

ninst = UBOUND(r, 1)
nbins = UBOUND(r, 2)

REDIM c(1 TO ninst)
REDIM d(1 TO ninst)

   FOR inst = 1 TO ninst

      d(inst) = 0
      FOR ibin = 1 TO nbins
         c(inst) = c(inst) + su(ibin) * r(inst, ibin) * dE(ibin)
      NEXT ibin

      d(inst) = v * b(inst) / c(inst) + (1 - v)

      IF d(inst) > MUM THEN d(inst) = MUM

   NEXT inst

   FOR ibin = 1 TO nbins

      RDsum = 0
      Rsum = 0
      FOR inst = 1 TO ninst

         IF inst <> n1 AND inst <> n2 THEN
            RDsum = RDsum + d(inst) * r(inst, ibin) / c(inst)
            Rsum = Rsum + r(inst, ibin) / c(inst)
         END IF

      NEXT inst
      su(ibin) = su(ibin) * RDsum / Rsum

   NEXT ibin

END SUB

SUB GenNoise (epsilon(), bexact(), b(), sig) STATIC
```

```
' ═══════════════════════════════════════════════════════════
' GenNoise simulates noise in the ideal signals by summings random numbers
' ═══════════════════════════════════════════════════════════

   ninst = UBOUND(b, 1)

   FOR inst = 1 TO ninst
      sumran = 0
      FOR jrand = 1 TO 12
         sumran = sumran + RND
      NEXT jrand
      epsilon(inst) = sig * (sumran - 6) * bexact(inst)
      b(inst) = (bexact(inst) + epsilon(inst))
   NEXT inst

END SUB

SUB GenRFs (sp(), e0(), e1(), r(), dE(), Yp()) STATIC

' ═══════════════════════════════════════════════════════════
' GenRFs inputs the k-edge energy for the first detector [e0(1)],
' calculates the k-edges for the rest of the detectors (for the fluorescers
' and the filters) based on e0(1), and calculates the response function,
' r(i,j).
' ═══════════════════════════════════════════════════════════

   SHARED SpaceType$, fedge, nopen, nclosed

   ninst = UBOUND(r, 1)
   nbins = UBOUND(r, 2)

   e0(1) = fedge
   IF e0(1) <= 0 THEN
      e0(1) = 1
      PRINT "The first k-edge can't be zero, it has been set to the default of 1"
      PRINT
   END IF
   PRINT

   IF UCASE$(LEFT$(SpaceType$, 1)) = "G" THEN
      FOR ibin = 1 TO nbins
         dE(ibin) = e0(1) * (64 ^ (ibin / nbins) - 64 ^ ((ibin - 1) / nbins))
      NEXT ibin

   ELSEIF UCASE$(LEFT$(SpaceType$, 1)) = "L" THEN
      FOR ibin = 1 TO nbins
         dE(ibin) = 63 * e0(1) / nbins
      NEXT ibin
   END IF

'  Calculate the k-edge of fluorescer for the rest of the detectors

   FOR inst = 2 TO nopen
      e0(inst) = SQR(2) ^ (inst - 1) * e0(1)
   NEXT inst
```

```
   FOR inst = nopen + 1 TO ninst
      e0(inst) = 2 ^ (inst - (nopen + 1)) * e0(1)
   NEXT inst

'  Calculate the k-edge of the filters

   FOR inst = 1 TO ninst
      e1(inst) = 2 * e0(inst)
   NEXT inst

'  Calculate the response functions from the detectors' k-edges

   FOR inst = 1 TO ninst
      e = e0(1)
      FOR ibin = 1 TO nbins
         e = e + dE(ibin) / 2
         IF e > e0(inst) THEN              'Above k-edge for detector i
            IF inst <= nopen THEN            'Closed response detectors
               IF e < e1(inst) THEN
                  r(inst, ibin) = (1 / e) * (1 - EXP(-2 * (e0(inst) / e) ^ 3))
                  r(inst, ibin) = r(inst, ibin) * EXP(-.25 * (e1(inst) / e) ^ 3)
               ELSE
                  r(inst, ibin) = (1 / e) * (1 - EXP(-2 * (e0(inst) / e) ^ 3))
                  r(inst, ibin) = r(inst, ibin) * EXP(-1.5 * (e1(inst) / e) ^ 3)
               END IF
            ELSE                          'Open response detectors
               r(inst, ibin) = (1 / e) * (1 - EXP(-3 * (e0(inst) / e) ^ 3))
            END IF
         ELSE                             'Below k-edge for detector i
            r(inst, ibin) = 0
         END IF

         LOCATE 15, 25
         PRINT USING "Initializing r(## ###)"; inst; ibin

         e = e + dE(ibin) / 2
      NEXT ibin
   NEXT inst

   CLS

'  Print the k-edges to screen and response functions to file

   CLS

'   OPEN "response.asc" FOR OUTPUT AS #1
'   FOR inst = 1 TO ninst
'      PRINT #1, inst
'      FOR ibins = 1 TO nbins
'         PRINT #1, r(inst, ibins)
'      NEXT ibins
```

```
'   NEXT inst
'   CLOSE #1
'   STOP

   FOR inst = 1 TO ninst

      Yp(inst) = 0

'     Integrating the predicted spectrum, Yp(inst)

      FOR ibin = 1 TO nbins
         Yp(inst) = Yp(inst) + sp(ibin) * r(inst, ibin) * dE(ibin)
      NEXT ibin

      FOR ibin = 1 TO nbins
         r(inst, ibin) = r(inst, ibin) / Yp(inst)
         NEXT ibin

   NEXT inst


END SUB

SUB HardCopy (index) STATIC

' ============================================================
' HardCopy produces hard copy output to printer
' ============================================================

   SHARED e0(), e1(), nopen, nclosed, bexact(), numinst
   SHARED b(), epsilon(), Q1(), Qloss(), iteropt, iteropt1
   SHARED inst, Chi, ChiExact, iter, fpred$, fexact$
   SHARED SetSmoother$, version$, datestring$, timestring$
   SHARED AbsNorm, RelNorm

   ninst = UBOUND(b, 1)
   niter = UBOUND(Q1, 2)

   IF index = 1 THEN

      LPRINT "Version;  "; version$
      LPRINT "Date:  "; datestring$
      LPRINT "Time:  "; timestring$
      LPRINT "Author:  Dennis Miller"
      LPRINT
      LPRINT "File for Exact Spectrum:  "; fexact$
      LPRINT "File for Predicted Spectrum:  "; fpred$
      LPRINT
      LPRINT "Smoother ON:  "; SetSmoother$
      LPRINT
```

```
      LPRINT
      LPRINT "inst    edge 1       edge 2    "
      FOR inst = 1 TO ninst
         IF inst <= nopen THEN
            LPRINT USING "##     ##.#####     ##.#####"; (inst); e0(inst); e1(inst)
         ELSE
            LPRINT USING "##     ##.#####     "; (inst); e0(inst)
         END IF
      NEXT inst
      LPRINT

      LPRINT " inst      bexact()       b()           epsilon()"
      FOR inst = 1 TO ninst
         LPRINT USING " ##      ##.###^^^^     ##.###^^^^     ##.###^^^^"; inst;
bexact(inst); b(inst); epsilon(inst)
      NEXT inst
      LPRINT

   ELSEIF index = 2 THEN

   ELSEIF index = 3 THEN

      LPRINT
      LPRINT "The optimum iteration number (minimization of Qloss): "; iteropt
      LPRINT

   ELSEIF index = 4 THEN

      LPRINT USING "##     ##    ##.####^^^^     ##.####^^^^     ##.####^^^^"; iter; inst;
Q1(inst, iter); Chi; ChiExact

   ELSEIF index = 5 THEN

      LPRINT USING "  0   ##.###^^^^   ##.###^^^^  ##.###^^^^   ##.###^^^^
##.###^^^^"; Chi; ChiExact; Qloss(0); AbsNorm; RelNorm

   ELSEIF index = 6 THEN

      LPRINT USING " ##   ##.###^^^^   ##.###^^^^  ##.###^^^^   ##.###^^^^
##.###^^^^"; iter; Chi; ChiExact; Qloss(iter); AbsNorm; RelNorm

   ELSEIF index = 7 THEN

      LPRINT "Iter   IO     Qterm          Chi^2          Chi^2 Exact"

   ELSEIF index = 8 THEN

      LPRINT USING "##     ##    ##.####^^^^     ##.####^^^^     ##.####^^^^"; iter; inst;
Q1(inst, iter); Chi; ChiExact

   ELSEIF index = 9 THEN

      LPRINT
      LPRINT "Iter   Chi^2        Chi^2 Exact  Qloss        AbsNorm       RelNorm"
```

```
   END IF

END SUB

SUB MeasToPred (b()) STATIC

' ============================================================
' MeasToPred calculates the measure-to-predicted ratio
' ============================================================

   SHARED sa(), sp(), r(), dE()

   ninst = UBOUND(r, 1)
   nbins = UBOUND(r, 2)

   FOR inst = 1 TO ninst

      b(inst) = 0
      FOR ibin = 1 TO nbins
         b(inst) = b(inst) + sa(ibin) * r(inst, ibin) * dE(ibin)
      NEXT ibin

   NEXT inst

END SUB

FUNCTION Qterm (Iomit, b(), su()) STATIC

' ============================================================
' Qterm generates the values of the Qloss function used as the test
' statistic in cross-validation
' ============================================================

   SHARED r(), dE(), sigma()

   ninst = UBOUND(r, 1)
   nbins = UBOUND(r, 2)

   REDIM c(1 TO ninst)

   sum = 0
   FOR ibin = 1 TO nbins
      sum = sum + r(Iomit, ibin) * su(ibin) * dE(ibin)
   NEXT ibin
   Qterm = ((sum - b(Iomit)) / sigma(Iomit)) ^ 2

END FUNCTION

SUB Setweights (SpaceType$, weight()) STATIC

' ============================================================
' Setweights generates the weighting matrix for the smoothing operation
' ============================================================
```

```
   SHARED start, ending, intave, pntwt, dE(), sdev

   nbins = UBOUND(dE, 1)

   Imax = INT((intave - 1) / 2)
   FOR i = 1 TO nbins

      FOR j = 1 TO nbins
         weight(i, j) = 0
      NEXT j

   NEXT i

'  Flat 3-bin averaging

   IF UCASE$(LEFT$(SpaceType$, 1)) = "G" OR UCASE$(LEFT$(SpaceType$, 1)) = "L" THEN

      jbin = 1
      FOR ibin = 1 TO nbins
         IF ibin > Imax AND ibin <= nbins - Imax THEN
            FOR jbin = ibin - Imax TO ibin + Imax
               weight(ibin, jbin) = 1 / intave
            NEXT jbin

         ELSEIF ibin <= Imax THEN
            FOR jbin = 1 TO ibin + Imax
               IF jbin <= Imax THEN
                  weight(ibin, jbin) = (2 / intave)
               ELSE
                  weight(ibin, jbin) = (1 / intave)
               END IF
            NEXT jbin

         ELSEIF ibin + Imax > nbins THEN
            FOR jbin = ibin - Imax TO nbins
               IF jbin >= ibin THEN
                  weight(ibin, jbin) = (2 / intave)
               ELSE
                  weight(ibin, jbin) = (1 / intave)
               END IF
            NEXT jbin
         END IF
      NEXT ibin
   END IF

END SUB

SUB Sinput STATIC

' ============================================================
' Sinput inputs all needed information
' ============================================================

CLS
```

```
SHARED start, ending, niter, intave, ninst, nbins, pntwt, sdev
SHARED SetNoise$, sig, fedge, SpaceType$, fexact$, fpred$, funfold$
SHARED nclosed, nopen, SetHardCopy$, fout1$
SHARED fout$, SetSmoother$, SetValidation$

ending = 64
intave = 3                  'number of E-bins averaged on (flat averaging)
nclosed = 6
niter = 10                  'number of iterations through Fit/Smooth
nbins = 120                 'number of energy intervals
nopen = 11
pntwt = .5
SetHardCopy$ = "N"
SetNoise$ = "N"             'random noise flag (Y=on, N=off)
SetSmoother$ = "Y"
SetValidation$ = "Y"
sig = .15                   'sigma
start = 1
sdev = .1

ninst = nopen + nclosed     'total number of instruments (detectors)

INPUT "Input the fluorescer's k-edge energy for the 1st detector: ", fedge
INPUT "Do you desire (L)inear or (G)eometric energy bins? ", SpaceType$
INPUT "Enter file name for actual spectrum: "; fexact$
INPUT "Enter file name for predicted spectrum: "; fpred$

funfold$ = fpred$     'set trial equal to prediction

INPUT "Enter file name for output spectrum: "; fout$
fout1$ = fout$ + ".out"

END SUB

SUB Smooth (su(), weight()) STATIC

' ============================================================
' Smooth provides 3-bin flat-averaging for the iterative unfold method
' ============================================================

   nbins = UBOUND(su, 1)

   REDIM temp(1 TO nbins)

   FOR ibin = 1 TO nbins
      temp(ibin) = 0
      Cum = 0
      FOR jbin = 1 TO nbins
         temp(ibin) = temp(ibin) + weight(ibin, jbin) * su(jbin)
      NEXT jbin
   NEXT ibin

   FOR ibin = 1 TO nbins
      su(ibin) = temp(ibin)
   NEXT ibin
```

```
END SUB

FUNCTION SpectralNorm (su(), ntype) STATIC

' ═══════════════════════════════════════════════════════════
' SpectralNorm calculates the norm of the unfolded spectrum (either absolute
' or relative dependent on ntype, 0 = absolute and 1 = relative)
' ═══════════════════════════════════════════════════════════

   SHARED sa()
   nbins = UBOUND(su, 1)

   sum = 0
   FOR ibin = 1 TO nbins
      IF ntype = 0 THEN denom = 1
      IF ntype = 1 THEN denom = sa(ibin)

      sum = sum + ((su(ibin) - sa(ibin)) / denom) ^ 2
   NEXT ibin

   SpectralNorm = sum / nbins

END FUNCTION

SUB Spectrum (filename$, s()) STATIC

' ═══════════════════════════════════════════════════════════
' Spectrum loads spectrum information for the given file
' ═══════════════════════════════════════════════════════════

   nbins = UBOUND(s, 1)

   OPEN filename$ FOR INPUT AS #2

   FOR ibin = 1 TO nbins
      INPUT #2, s(ibin)
   NEXT ibin
   CLOSE #2

END SUB
```

Appendix B: Additional Noise Set for Test Case 1

This appendix is a continuation of the results of Section IV. The inputs are identical to the series of test cases prefixed TC1 in Section IV, except the random number generator in CVIter.bas was reseeded to generated a different noise set. The following are the results of changing noise effects case TC1.

Table 17: Case TC1.0 (smoother off, 2nd noise set)

Iter	χ^2	χ^2_s	Q_l	L_a	L_r
0	7.0760E+03	6.9083E+03	7.0760E+03	8.1536E-05	1.8260E+02
1	3.7144E+01	1.2463E+01	1.1581E+02	1.1935E-05	1.6000E+00
2	1.9649E+01	2.5925E+00	6.7144E+01	8.0145E-06	8.9513E-01
3	1.8108E+01	3.0558E+00	6.6892E+01	8.8215E-06	7.1876E-01
4	1.6779E+01	3.3035E+00	6.7789E+01	1.0172E-05	5.7400E-01
5	1.5622E+01	3.5494E+00	6.8486E+01	1.1978E-05	4.5841E-01
6	1.4613E+01	3.8155E+00	6.9024E+01	1.4093E-05	3.6813E-01
7	1.3726E+01	4.0973E+00	6.9465E+01	1.6414E-05	2.9858E-01
8	1.2940E+01	4.3894E+00	6.9845E+01	1.8873E-05	2.4575E-01
9	1.2239E+01	4.6873E+00	7.0184E+01	2.1426E-05	2.0632E-01
10	1.1610E+01	4.9879E+00	7.0489E+01	2.4043E-05	1.7764E-01

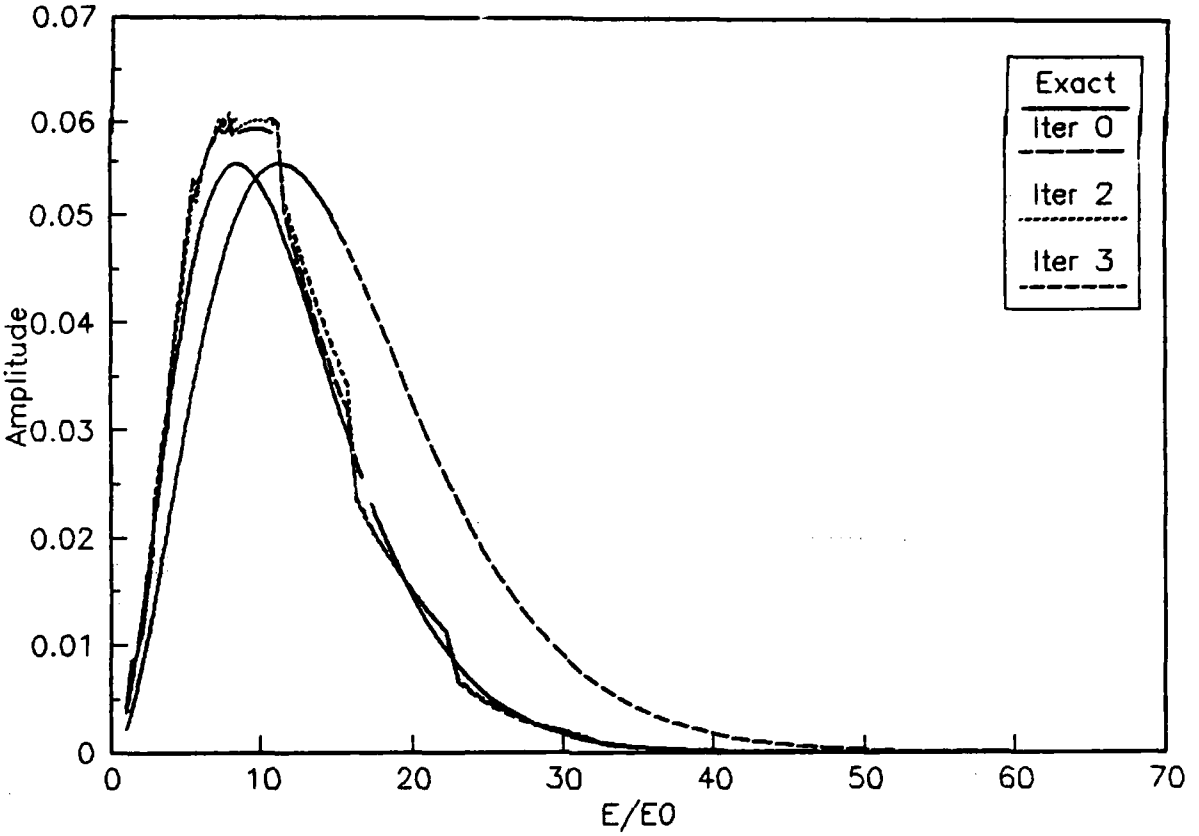


Figure 19: Case TC1.0 (smoother off, 2nd noise set)

Table 18: Case TC1.1 (smoother off, 2nd noise set)

Iter	χ^2	χ^2_i	Q_i	L_a	L_r
0	2.0275E+01	0.0000E+00	2.0275E+01	0.0000E+00	0.0000E+00
1	1.8870E+01	2.3531E+00	3.7177E+01	5.2818E-06	5.1626E-03
2	1.7235E+01	2.8519E+00	4.0338E+01	7.6921E-06	1.0107E-02
3	1.5897E+01	3.1316E+00	4.2052E+01	9.8909E-06	1.6739E-02
4	1.4769E+01	3.3948E+00	4.3392E+01	1.2157E-05	2.4941E-02
5	1.3798E+01	3.6713E+00	4.4605E+01	1.4533E-05	3.4387E-02
6	1.2950E+01	3.9617E+00	4.5759E+01	1.7008E-05	4.4784E-02
7	1.2201E+01	4.2621E+00	4.6874E+01	1.9564E-05	5.5894E-02
8	1.1534E+01	4.5685E+00	4.7951E+01	2.2182E-05	6.7532E-02
9	1.0939E+01	4.8778E+00	4.8987E+01	2.4846E-05	7.9547E-02
10	1.0404E+01	5.1873E+00	4.9981E+01	2.7544E-05	9.1824E-02

Table 19: Case TC1.0 (smoother on, 2nd noise set)

Iter	χ^2	χ^2_s	Q_l	L_a	L_r
0	7.0760E+03	6.9083E+03	7.0760E+03	8.1536E-05	1.8260E+02
1	4.3853E+01	1.8059E+01	1.2509E+02	1.124[illegible]-05	1.9583E+00
2	1.9702E+01	2.4934E+00	6.4496E+01	7.1811E-06	1.0853E+00
3	1.8202E+01	2.9071E+00	6.3251E+01	8.0396E-06	9.7822E-01
4	1.6955E+01	3.1155E+00	6.3535E+01	9.2711E-06	8.7552E-01
5	1.5949E+01	3.3252E+00	6.3997E+01	1.0767E-05	7.8540E-01
6	1.5101E+01	3.5439E+00	6.4436E+01	1.2395E-05	7.0456E-01
7	1.4387E+01	3.7690E+00	6.4885E+01	1.4074E-05	6.3308E-01
8	1.3775E+01	3.9946E+00	6.5334E+01	1.5753E-05	5.6987E-01
9	1.3249E+01	4.2172E+00	6.5782E+01	1.7404E-05	5.1412E-01
10	1.2792E+01	4.4343E+00	6.6227E+01	1.9009E-05	4.6504E-01

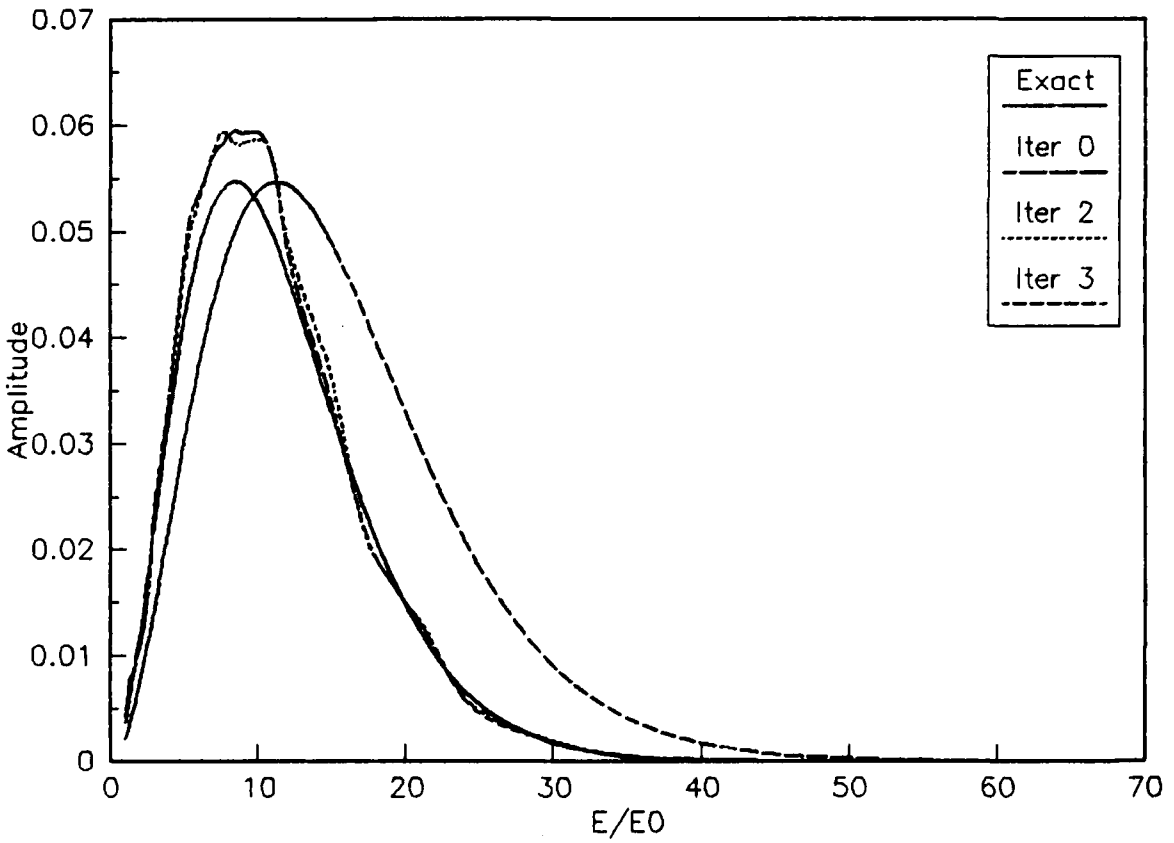


Figure 20: Case TC1.0 (smoother on, 2nd noise set)

Table 20: Case TC1.1 (smoother on, 2nd noise set)

Iter	χ^2	χ^2_i	Q_i	L_a	L_r
0	2.0275E+01	0.0000E+00	2.0275E+01	0.0000E+00	0.0000E+00
1	1.9345E+01	2.2474E+00	3.7105E+01	5.1385E-06	4.7561E-03
2	1.7918E+01	2.6997E+00	4.1091E+01	7.3144E-06	8.5828E-03
3	1.6778E+01	2.9449E+00	4.3321E+01	9.1606E-06	1.3273E-02
4	1.5842E+01	3.1652E+00	4.5142E+01	1.0923E-05	1.8770E-02
5	1.5053E+01	3.3866E+00	4.6756E+01	1.2650E-05	2.4856E-02
6	1.4379E+01	3.6117E+00	4.8267E+01	1.4346E-05	3.1330E-02
7	1.3798E+01	3.8376E+00	4.9697E+01	1.6004E-05	3.8030E-02
8	1.3293E+01	4.0614E+00	5.1057E+01	1.7614E-05	4.4830E-02
9	1.2851E+01	4.2809E+00	5.2351E+01	1.9172E-05	5.1635E-02
10	1.2464E+01	4.4944E+00	5.3580E+01	2.0672E-05	5.8370E-02

Appendix C: Additional Noise Set for Test Case 2

This appendix is a continuation of the results of Section IV. The inputs are identical to the series of test cases prefixed TC2 in Section IV, except the random number generator in CVIter.bas was reseeded to generated a different noise set. The following are the results of changing noise effects case TC2.

Table 21: Case TC2.0 (smoother off, 2nd noise set)

Iter	χ^2	χ^2_s	Q_t	L_a	L_r
0	2.8788E+02	3.1522E+02	2.8788E+02	4.7672E-03	6.1341E-01
1	8.8249E+01	9.0178E+01	1.3938E+02	1.8998E-03	2.9801E-01
2	1.7637E+01	1.2828E+01	4.6268E+01	4.7876E-04	1.4517E-01
3	1.3306E+01	7.3181E+00	4.1795E+01	3.6569E-04	1.3704E-01
4	1.1662E+01	6.3850E+00	4.2463E+01	3.8398E-04	1.3977E-01
5	1.0430E+01	6.3304E+00	4.3731E+01	4.3777E-04	1.4510E-01
6	9.4243E+00	6.5598E+00	4.5096E+01	5.0891E-04	1.5174E-01
7	8.5954E+00	6.9103E+00	4.6469E+01	5.9049E-04	1.5924E-01
8	7.9074E+00	7.3177E+00	4.7818E+01	6.7866E-04	1.6731E-01
9	7.3321E+00	7.7493E+00	4.9128E+01	7.7097E-04	1.7575E-01
10	6.8470E+00	8.1860E+00	5.0388E+01	8.6574E-04	1.8441E-01

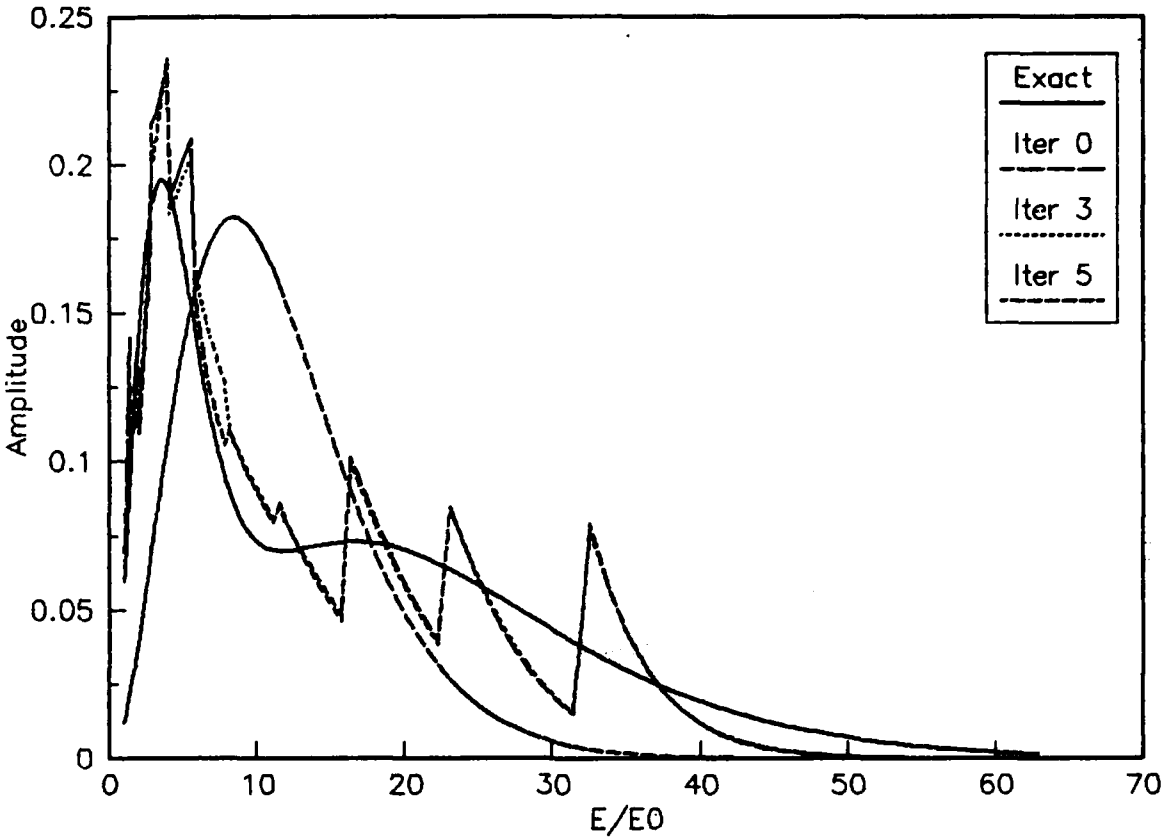


Figure 21: Case TC2.0 (smoother off, 2nd noise set)

Table 22: Case TC2.1 (smoother off, 2nd noise set)

Iter	χ^2	χ^2_i	Q_i	L_a	L_r
0	1.8445E+02	1.8141E+02	1.8445E+02	2.6977E-03	3.3271E-01
1	2.6800E+01	1.9368E+01	5.0118E+01	4.9980E-04	5.2829E-02
2	1.6419E+01	6.0983E+00	4.1733E+01	2.2261E-04	2.6409E-02
3	1.4374E+01	4.4238E+00	4.3895E+01	2.1956E-04	2.7689E-02
4	1.2926E+01	4.2343E+00	4.6107E+01	2.6737E-04	3.3609E-02
5	1.1721E+01	4.4402E+00	4.8085E+01	3.3612E-04	4.1290E-02
6	1.0711E+01	4.7997E+00	4.9928E+01	4.1689E-04	5.0008E-02
7	9.8611E+00	5.2288E+00	5.1679E+01	5.0523E-04	5.9423E-02
8	9.1402E+00	5.6890E+00	5.3345E+01	5.9838E-04	6.9307E-02
9	8.5242E+00	6.1589E+00	5.4927E+01	6.9452E-04	7.9492E-02
10	7.9935E+00	6.6253E+00	5.6423E+01	7.9239E-04	8.9852E-02

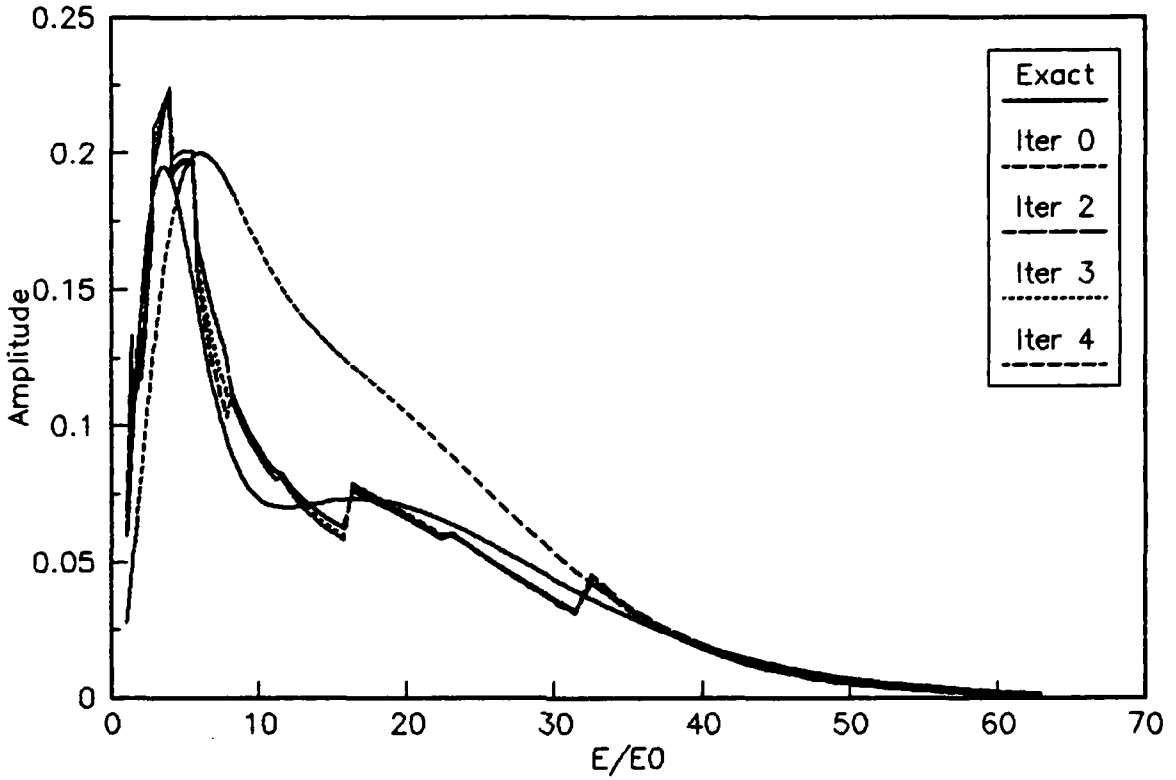


Figure 22: Case TC2.1 (smoother off, 2nd noise set)

Table 23: Case TC2.2 (smoother off, 2nd noise set)

Iter	χ^2	χ_i^2	Q_i	L_a	L_r
0	2.0275E+01	0.0000E+00	2.0275E+01	0.0000E+00	0.0000E+00
1	1.7837E+01	2.0967E+00	3.5241E+01	7.4127E-05	5.1165E-03
2	1.5678E+01	2.7595E+00	3.9461E+01	1.2750E-04	1.0021E-02
3	1.4073E+01	3.2518E+00	4.2299E+01	1.9142E-04	1.6127E-02
4	1.2764E+01	3.7105E+00	4.4647E+01	2.6530E-04	2.3385E-02
5	1.1664E+01	4.1735E+00	4.6748E+01	3.4723E-04	3.1588E-02
6	1.0730E+01	4.6474E+00	4.8691E+01	4.3525E-04	4.0521E-02
7	9.9308E+00	5.1282E+00	5.0510E+01	5.2766E-04	4.9994E-02
8	9.2424E+00	5.6095E+00	5.2219E+01	6.2310E-04	5.9856E-02
9	8.6458E+00	6.0849E+00	5.3824E+01	7.2053E-04	6.9981E-02
10	8.1252E+00	6.5490E+00	5.5330E+01	8.1914E-04	8.0273E-02

Table 24: Case TC2.0 (smoother on, 2nd noise set)

Iter	χ^2	χ^2_I	Q_i	L_a	L_r
0	2.8788E+02	3.1522E+02	2.8788E+02	4.7672E-03	6.1341E-01
1	8.9677E+01	9.1671E+01	1.4083E+02	1.8440E-03	2.7991E-01
2	1.8301E+01	1.2698E+01	4.5893E+01	4.0690E-04	1.2116E-01
3	1.3975E+01	6.7272E+00	4.1036E+01	2.7637E-04	1.0692E-01
4	1.2518E+01	5.5320E+00	4.1639E+01	2.7873E-04	1.0451E-01
5	1.1485E+01	5.2783E+00	4.2936E+01	3.1469E-04	1.0500E-01
6	1.0665E+01	5.3157E+00	4.4363E+01	3.6484E-04	1.0688E-01
7	1.0007E+01	5.4771E+00	4.5819E+01	4.2176E-04	1.0957E-01
8	9.4758E+00	5.6951E+00	4.7270E+01	4.8152E-04	1.1273E-01
9	9.0439E+00	5.9378E+00	4.8699E+01	5.4174E-04	1.1613E-01
10	8.6905E+00	6.1874E+00	5.0093E+01	6.0091E-04	1.1963E-01

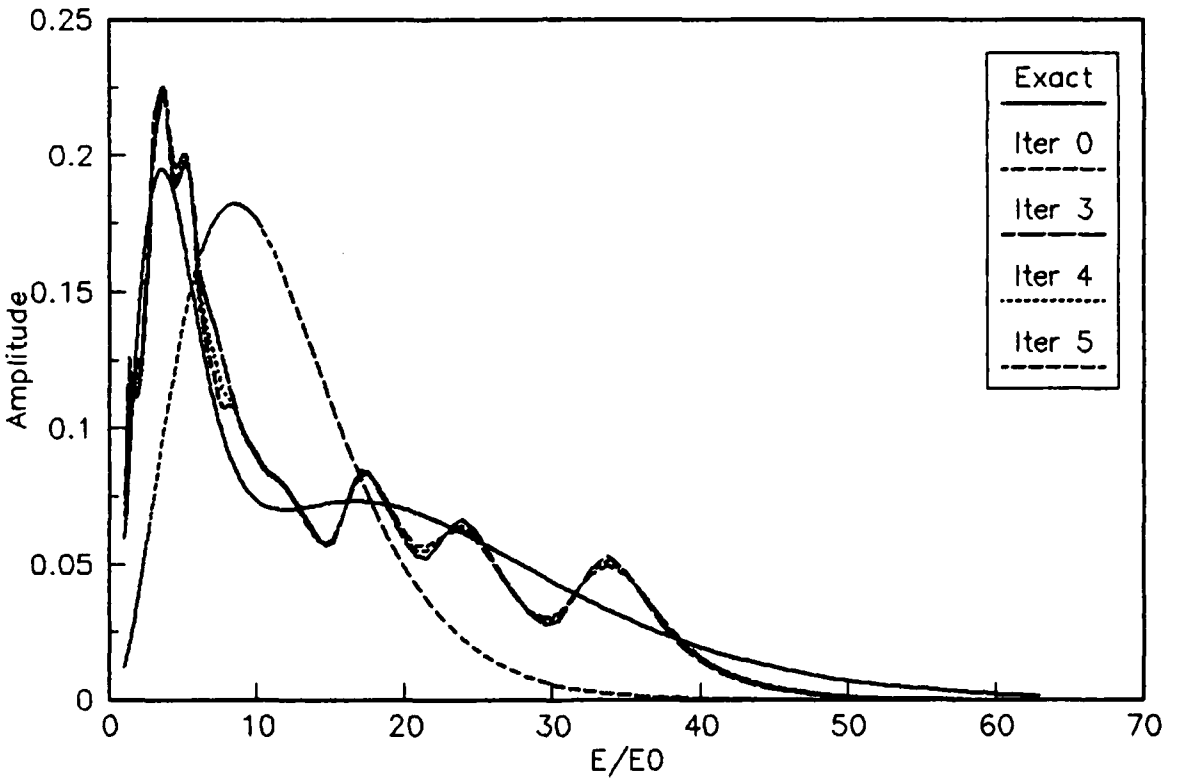


Figure 23: Case TC2.0 (smoother on, 2nd noise set)

Table 25: Case TC2.1 (smoother on, 2nd noise set)

Iter	χ^2	χ^2_f	Q_l	L_a	L_r
0	1.8445E+02	1.8141E+02	1.8445E+02	2.6977E-03	3.3271E-01
1	2.7089E+01	1.9662E+01	4.9971E+01	4.9271E-04	5.1342E-02
2	1.6727E+01	6.1370E+00	4.1481E+01	2.0696E-04	2.3084E-02
3	1.4866E+01	4.3759E+00	4.3712E+01	1.9564E-04	2.2610E-02
4	1.3618E+01	4.0998E+00	4.6022E+01	2.3208E-04	2.6427E-02
5	1.2603E+01	4.2064E+00	4.8085E+01	2.8524E-04	3.1611E-02
6	1.1772E+01	4.4498E+00	5.0007E+01	3.4589E-04	3.7430E-02
7	1.1088E+01	4.7487E+00	5.1831E+01	4.0963E-04	4.3555E-02
8	1.0521E+01	5.0669E+00	5.3572E+01	4.7392E-04	4.9785E-02
9	1.0048E+01	5.3861E+00	5.5232E+01	5.3715E-04	5.5978E-02
10	9.6497E+00	5.6959E+00	5.6813E+01	5.9831E-04	6.2037E-02

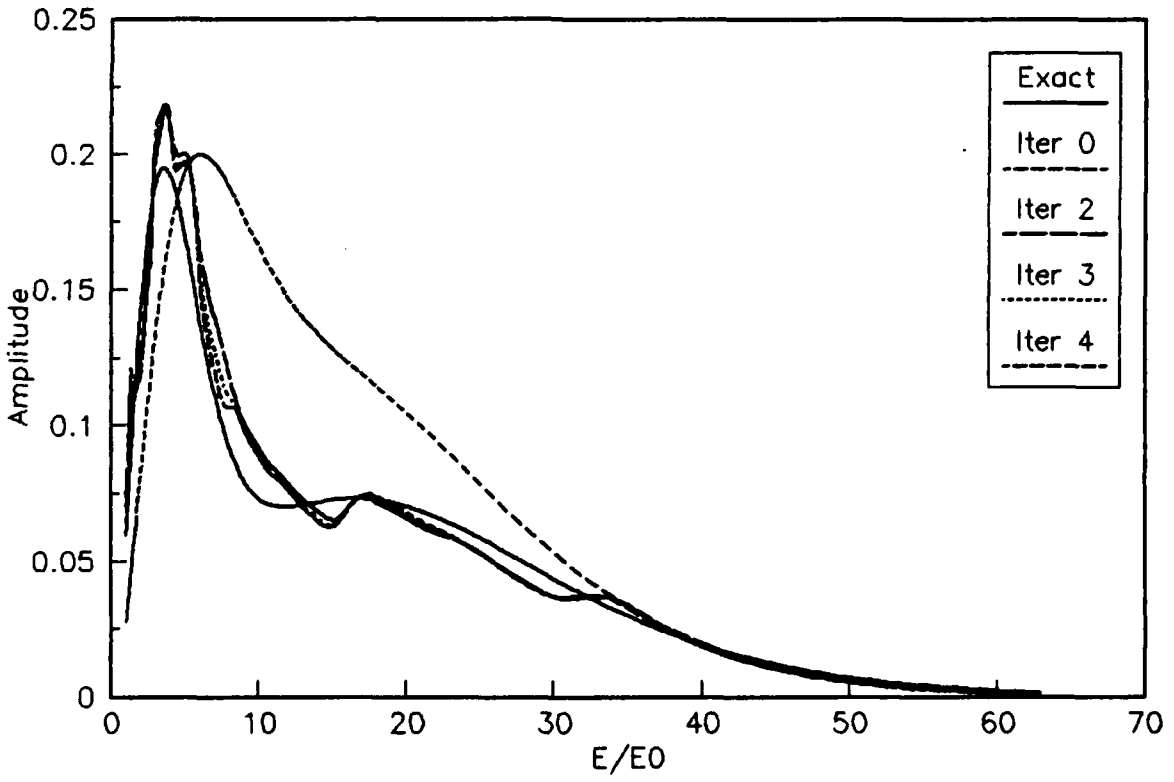


Figure 24: Case TC2.1 (smoother on, 2nd noise set)

Table 26: Case TC2.2 (smoother on, 2nd noise set)

Iter	χ^2	χ^2_s	Q_l	L_a	L_r
0	2.0275E+01	0.0000E+00	2.0275E+01	0.0000E+00	0.0000E+00
1	1.7987E+01	2.0766E+00	3.5034E+01	7.1943E-05	4.9554E-03
2	1.5999E+01	2.7191E+00	3.9309E+01	1.2005E-04	9.3123E-03
3	1.4581E+01	3.1711E+00	4.2260E+01	1.7466E-04	1.4384E-02
4	1.3459E+01	3.5682E+00	4.4750E+01	2.3481E-04	2.0080E-02
5	1.2540E+01	3.9481E+00	4.6992E+01	2.9846E-04	2.6204E-02
6	1.1778E+01	4.3200E+00	4.9073E+01	3.6372E-04	3.2566E-02
7	1.1140E+01	4.6830E+00	5.1026E+01	4.2903E-04	3.9014E-02
8	1.0603E+01	5.0335E+00	5.2869E+01	4.9325E-04	4.5431E-02
9	1.0148E+01	5.3682E+00	5.4611E+01	5.5552E-04	5.1728E-02
10	9.7600E+00	5.6845E+00	5.6258E+01	6.1530E-04	5.7842E-02

Appendix D: Pseudo Code for CVIter.bas

Pseudo Code for Main Program

```
    Initialize all necessary variables

    Call Spectrum (read from file the unfolded, predicted and
exact spectra)

    Call GenRFs (generate response data from equations 3 and
4)

    Call Setweights (establish weighting matrix for smoothing
operation)

    Call MeasToPred (calculate b_i for each instrument)

    If SetNoise$ = "Y"

        Call GenNoise (simulate noise and add to ratio as in
equation 23)

    End if

    If SetValidation$ = "Y"    (start cross-validation)

        Omit instrument

        Initialize su to sguess  (set unfolded equal to
guessed spectrum)

        Calculate statistics for 0th iteration (guessed spec-
trum)

        Fit unfolded spectrum (1 instrument omitted)

        Smooth unfolded spectrum  (if implemented, 1 instru-
ment omitted)

        Calculate test statistics (except Q1 is a summation
over all
           instruments omitted)

        Repeat fit and smooth for each iteration and tabulate
statistics

        Repeat omitting next instrument

    End if

    Calculate Q_i as in equation 22 (using subset of instru-
ments)
```

Reinitialize guessed spectrum

Fit unfolded spectrum (using all instruments)

Smooth unfolded spectrum (if implemented, using all instruments)

Repeat for each iteration

Bibliography

1. Daniel, Capt. Russell B. An Approximation Technique for Solving a System of Fredholm Integral Equations for Asymmetric Detector Response Functions. MS Thesis AFIT/GNE/ENG/88M-3. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH. March 1988.

2. Carter, Capt. Michael E. Comparison of Basis Function and Iteration Solutions to the Fredholm Integral Equation of the First Kind. MS Thesis AFIT/GNE/ENP/89M-1. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH. March 1989.

3. Stone, M. "Cross-Validatory Choice and Assessment of Statistical Predictions," Journal of the Royal Statistical Society, Ser. B, 2: 111-147 (1974)

4. Geisser, Seymour. "The Predictive Sample Reuse Method with Application," Journal of the American Statistical Association, 258: 320-328 (June 1975)

5. Mathews, K.A., Air Force Institute of Technology. Private Communications. (1989)

6. Bridgman, Charles J., Class Notes in NENG 631, Prompt Effects of Nuclear Weapons. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 1989.

7. Larimore, Wallace E. and Raman K. Mehra. "The Problem of Overfitting Data," Byte, (October 1985)

Vita

Lieutenant Dennis J. Miller was born on 17 September 1957 in Seattle, Washington. He graduated from A. C. Davis Senior High School in 1975. He enlisted in the United States Air Force in 1976. After basic training, he was assigned to the 410th Bombardment Wing, K. I. Sawyer AFB, Michigan and then at the 92nd Bombardment Wing, Fairchild AFB, Washington, where he served as an aircraft crew chief on B-52 G/H aircraft. He was accepted to the Airman Education and Commissioning Program (AECP) in 1983 and received a Bachelor of Science Degree in Engineering (Nuclear Emphasis) from the University of Washington in 1985. He was commissioned a Second Lieutenant in the Air Force in the spring of 1986. After commissioning, he was assigned to Air Force Logistics Command (AFLC), Tinker AFB, Oklahoma, where he was assigned as a Survivability/Vulnerabilty Officer. He entered the Nuclear Engineering program at the Air Force Institute of Technology in August of 1988.

Permanent Address: 1216 Rock Ave.
Yakima, WA 98902

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GNE/ENP/90M-5	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Engineering 6b. OFFICE SYMBOL (If applicable) AFIT/ENP	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB OH 45433-6583	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS N/A PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
THE FEASIBILITY OF CROSS-VALIDATION AS A PARAMETER PREDICTOR FOR THE ITERATIVE UNFOLD TECHNIQUE

12. PERSONAL AUTHOR(S)
Dennis J. Miller, B.S., 1st Lt, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
MS Thesis	FROM _____ TO _____	1990 March	117

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Cross-Validation Deconvolution Function Fitting
18	4		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Thesis Adivisor: Kirk A. Mathews, LCDR, USN
Assistant Professor
Department of Engineering

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Kirk A. Mathews, LCDR, USN	22b. TELEPHONE (Include Area Code) (513) 255-4489 22c. OFFICE SYMBOL AFIT/ENP

The iterative unfold technique provides a means for approximating a solution to a system of Fredholm integral equations. This method consists of modifying a guess spectrum through a fit process using data collected from experiment. The unfolded spectrum is then smoothed to reduce undesirable artifacts that result from the fitting process. Finally, the entire iterative process is then repeated as necessary to provide an approximation to the exact incident spectrum. Presently, the iterative unfold method lacks an independent measure of how well the unfolded spectrum approximates the exact spectrum. Consequently, user judgement is necessary, resulting in possible data overfitting. Cross-validation is a method which selectively partitions the measured data into subsets. The subsets of data are used to predict omitted data. A cross-validatory loss statistic can be formulated and minimized to predict the optimum stopping point for the iterative unfold process.

The iterative unfold technique was implemented into a computer program. Cross-validation was introduced to determine the cross-validatory loss statistic. The loss statistic was minimized by varying the number of iterations of the unfold routine over a wide range. Test cases were developed to compare the cross-validatory loss statistic with a χ^2 formulated using ideal signals.

Two basic test cases were studied. In these two cases, the guessed spectrum, the simulated noise set, and implementation of the iterative unfold technique were varied. Cross-validation provided an estimate for the stopping point of the unfold routine in each case. Using the smooth algorithm in tandem with the fit algorithm produced an overall better fit than using the fit algorithm alone. When the guessed spectrum was set equal to the exact spectrum; cross-validation predicted no modification to the guessed spectrum, while judgement alone tended to overfit data. Stoppage based on the value of a χ^2 test statistic proved highly problem dependent. In contrast, cross-validation averaged within one iteration of the optimum regardless of the problem parameters.